

VisualAge Pacbase



Guide d'installation Serveur Windows 2000 ou NT & Composants Client

Version 3.0



VisualAge Pacbase



Guide d'installation Serveur Windows 2000 ou NT & Composants Client

Version 3.0

Note

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Notices» à la page v.

En application de votre contrat de licence, vous pouvez consulter ou télécharger la documentation de VisualAge Pacbase, régulièrement mise à jour, à partir de :

http://www.ibm.com/software/ad/vapacbase/productinfo_f.htm

La section Catalogue dans la page d'accueil de la Documentation vous permet d'identifier la dernière édition disponible du présent document.

Troisième édition (mai 2002)

La présente édition s'applique à :

- VisualAge Pacbase Version 3.0

Vous pouvez nous adresser tout commentaire sur ce document (en indiquant sa référence) via le site Web de notre Support Technique à l'adresse suivante : <http://www.ibm.com/software/ad/vapacbase/support.htm> ou en nous adressant un courrier à :

IBM Paris Laboratory
1, place Jean-Baptiste Clément
93881 Noisy-le-Grand, France.

IBM pourra disposer comme elle l'entendra des informations contenues dans vos commentaires, sans aucune obligation de sa part.

© Copyright International Business Machines Corporation 1983,2002. All rights reserved.

Table des matières

Notices	v	Le fichier bases.ini	27
Marques	vii	Le fichier vaplocat.ini	29
Chapitre 1. Préambule	1	Désinstallation de composants Client.	31
Objet du manuel	1	Chapitre 5. Tests	33
Architecture du produit	1	Liste des utilitaires principaux	33
Contenu de la livraison	2	Tests de l'installation	33
Bibliographie	2	Tests de mise à jour TP/batch ,	
Chapitre 2. Prérequis	3	Edition-Génération.	34
Prérequis partie Serveur	3	Tests des procédures de la base	
Matériel et logiciel	3	Administration	34
Espace disque.	3	Tests des procédures de la base de	
Prérequis partie Client	3	Développement	34
Matériel.	3	Tests des utilitaires d'extraction	35
Espace disque.	4	Chapitre 6. Réinstallation Serveur	37
Logiciel	4	Réinstallation	37
Communications	4	Désinstallation	37
Chapitre 3. Installation de la partie Serveur 5		Suppression des données.	37
Installation	5	Chapitre 7. Reprises	39
Installation du Système.	5	Reprises des bases des versions 2.0 et 2.5	39
Installation du Référentiel	6	Opérations à effectuer.	39
Base Administration.	6	Reprise des paramètres utilisateur (PE25)	41
Base de Développement	6	PE25 - Présentation générale	41
Créer nouvelle Base	7	PE25 - Entrées / Traitements / Résultats	42
Paramètres de Configuration	7	PE25 - Description des étapes	43
Liste des dates des composants	7	PE25 - Script d'exécution.	44
INSL - Script d'exécution	7	Reprise de la base de Développement (PC25)	46
Chapitre 4. Installation/Ré-installation des composants Client	11	PC25 - Présentation générale	46
Informations préalables	11	PC25 - Remarques sur les données reprises	46
Principes et architectures de la communication	12	PC25 - Entrées / Traitements / Résultats	48
Lancement de l'installation	14	PC25 - Description des étapes	52
Administrator & Developer workbench	14	PC25 - Script d'exécution	52
Outils eBusiness	17	Reprise commandes d'édition-génération (PG20)	55
Station de Travail VisualAge Pacbase.	23	PG20 - Présentation générale	55
Pacbase Web Connection.	23	PG20 - Entrées / Traitements / Résultats	56
Middleware	25	PG20 - Description des étapes	59
Mise à jour des paramètres de la communication	27	PG20 - Script d'exécution	60
		Reprise commandes d'édition-génération (PG25)	64
		PG25 - Présentation générale	64
		PG25 - Entrées / Traitements / Résultats	65
		PG25 - Description des étapes	68

PG25 - Script d'exécution	69	Composants de la partie Serveur	95
Reprise PEI (PP25)	72	Présentation générale	95
PP25 - Présentation générale	72	Documentation en ligne	95
PP25 - Entrées / Traitements / Résultats	73	Squelettes de génération	95
PP25 - Description des étapes	74	Base administration	95
PP25 - Script d'exécution.	75	Fichiers de la Base administration.	95
Reprise paramètres Pac/Transfer (UV25)	77	Sauvegarde de la Base administration	96
UV25 - Présentation générale	77	Base de développement	96
UV25 - Entrées / Traitements / Résultats	77	Fichiers de la Base de développement	96
UV25 - Description des étapes	78	Fichiers de sauvegarde Base de	
UV25 - Script d'exécution	80	développement	97
Reprise mouvements MB (MB25)	81	Modules - fichiers spécifiques	97
MB25 - Présentation générale	81	Pac/Impact	97
MB25 - Description des étapes	82	DSMS	97
MB25 - Script d'exécution	82	PAF.	98
Reprise mouvements GY (GY25)	83	Bibliothèques et fichiers	
GY25 - Présentation générale	83	complémentaires	98
GY25 - Description des étapes	84	Chapitre 9. Annexes	99
GY25 - Script d'exécution	84	Installation du Modèle de la Base	
Reprise mouvements MB (MB30)	85	administration	99
MB30 - Présentation générale	85	VINS pour Administration - Présentation	
MB30 - Description des étapes	86	générale	99
MB30 - Script d'exécution	86	VINS - Entrées / Traitements / Résultats	99
Reprise mouvements GY (GY30)	87	VINS - Description des étapes	100
GY30 - Présentation générale	87	VINS - Script d'exécution	101
GY30 - Description des étapes	88	Installation du Modèle de la Base de	
GY30 - Script d'exécution	88	développement	103
Reprise mouvements PJ (PJ25)	89	VINS - Présentation générale	103
PJ25 - Présentation générale.	89	VINS - Entrées / Traitements / Résultats	104
PJ25 - Description des étapes	90	VINS - Description des étapes	105
PJ25 - Script d'exécution	90	VINS - Script d'exécution	106
Procédures - Tableau récapitulatif des			
changements	92		
Chapitre 8. Les composants	95		

Notices

Ce document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM. Cela ne signifie pas qu'IBM ait l'intention de les annoncer dans tous les pays où la compagnie est présente. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante : IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk NY 10504-1785, U.S.A.

Les détenteurs de licences du présent produit souhaitant obtenir des informations sur celui-ci à des fins : (i) d'échange d'informations entre des programmes développés indépendamment et d'autres programmes (y compris celui-ci) et (ii) d'utilisation mutuelle des informations ainsi échangées doivent s'adresser à : IBM Paris Laboratory, SMC Department, 1 place J.B.Clément, 93881 Noisy-Le-Grand Cedex, France. De telles informations peuvent être mises à la disposition du Client et seront soumises aux termes et conditions appropriés, y compris dans certains cas au paiement d'une redevance.

IBM peut modifier ce document, le produit qu'il décrit ou les deux.

Marques

IBM est une marque d'International Business Machines Corporation, Inc. AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, OS/2, VisualAge Pacbase, RACF, RS/6000, SQL/DS et VisualAge sont des marques d'International Business Machines Corporation, Inc. dans certains pays.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation dans certains pays.

UNIX est une marque enregistrée aux Etats-Unis et/ou dans d'autres pays et utilisée avec l'autorisation exclusive de la société X/Open Company Limited.

D'autres sociétés peuvent être propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

Chapitre 1. Préambule

Objet du manuel

L'objet de ce manuel est de guider l'administrateur dans l'installation du produit pour :

- la partie Serveur,
- la partie Client,
- la communication.

Lorsque l'installation est complète, il est conseillé de procéder à des tests à l'aide du jeu d'essai fourni sur le support d'installation.

Vous trouverez également une description des opérations à effectuer dans le cas de l'installation d'une version de correction.

Architecture du produit

VisualAge Pacbase est un outil de conception, de développement et de maintenance d'applications eBusiness de type graphique (GUI) ou mode caractère (TUI), Web, conversationnel ou batch.

Cet outil est constitué :

- d'une partie Serveur (incluant l'interface mode caractère).
- d'une partie Client (interface graphique)

Ces deux parties communiquent entre elles grâce à un middleware encapsulé et fourni par IBM.

Note : l'utilisation des écrans en mode caractère reste possible pour certaines fonctionnalités.

Vous trouverez une description détaillée des composants Serveur dans le chapitre "Les composants".

La partie Serveur

Elle est constituée des éléments suivants :

- les éléments du système : programmes, fichiers (dont documentation en ligne) et paramètres,

- la Base Administration : contenant les paramètres utilisateur et autres paramètres,
- une (ou plusieurs) Base(s) de Développement.

La partie Client

Elle est constituée des composants suivants :

- Administrator workbench,
- Developer workbench qui inclut les modules Batch et eBusiness (qui peuvent être installés indépendamment l'un de l'autre),
- La Station de Travail VisualAge Pacbase,
- Les outils eBusiness,
- Pacbase Web Connection.

Les fonctions de communication permettent de gérer les échanges entre les parties Serveur et Client du produit en utilisant, selon l'environnement, les principaux protocoles de communication du marché.

Contenu de la livraison

Le contenu d'une livraison du produit varie selon les termes de votre commande :

- Guide d'Installation,
- CD-Rom ou cartouche, selon l'environnement, pour l'installation de la partie serveur,
- CD-ROM Workstation Components,
- CD-ROM VisualAge Java,
- CD-ROM de documentation du produit.

Bibliographie

Les procédures, utilisées par l'Administrateur pour la gestion des Bases et l'administration des versions, et les utilitaires fournis, sont documentés dans le Guide des Procédures de l'Administrateur.

La gestion des paramètres utilisateur (mise à jour des clés d'accès, des codes utilisateur et des autorisations d'accès) est documentée dans l'aide en ligne de Administrator workbench.

Les informations concernant la communication des contrôles entre le système de sécurité éventuellement installé sur le site et VisualAge Pacbase (contrôle des codes utilisateur, des mots de passe et des autorisations d'accès) sont documentées dans le manuel "Interfaces Systèmes de Sécurité".

Chapitre 2. Prérequis

Prérequis partie Serveur

Matériel et logiciel

- Architecture : Un serveur Windows/NT ou Windows 2000 et des stations de travail Windows.
- Processeur : Poste supportant soit Windows NT server (4.0 minimum) soit Windows 2000.
- Mémoire : 96 Mo. La mémoire nécessaire peut augmenter en fonction du nombre de serveurs installés sur la même machine.
- Logiciels :
 - Microsoft Windows Script, version 5.1 minimum. Vous pouvez la télécharger à partir de l'url suivante :
www.microsoft.com/msdownload/vbscript/scripting.asp
- Micro Focus Application Server.
- Support d'installation : Lecteur de CD-ROM

Espace disque

L'espace disque occupé par les fichiers dépend de l'importance des applications gérées par le système.

Le tableau ci-dessous indique approximativement (en millions d'octets) l'espace disque nécessaire à l'installation des serveurs :

Encombrement installation	Total
Total installation	95
Total Système	59
Total fichiers Utilisateur de tests d'installation	35
Total fichiers Administrateur de tests d'installation	260 ko

Prérequis partie Client

Matériel

Les caractéristiques matérielles nécessaires à l'installation des composants clients VisualAge Pacbase sont les suivantes :

- Processeur : Intel Pentium III 450 Mhz minimum ou processeur compatible.

- Moniteur : Moniteur graphique (800x600) VGA ou plus grande résolution (XGA ou SVGA).
- Lecteur de CD-ROM.
- Carte : carte adaptée au réseau du site.
- Mémoire (RAM): 128 Mo (256 Mo conseillé).
- Logiciel: Microsoft Windows Script Host (Version 5.1 au minimum).

Espace disque

Espace disque nécessaire :

- 58 Mo pour Administrator & Developer workbench.
- 15 Mo pour la Station de Travail VisualAge Pacbase.

Logiciel

Les composants Client de VisualAge Pacbase requièrent que votre poste soit équipé d'un Windows 32 bits, au choix :

- Windows 98,
- Windows/NT version 4.0 avec Service Pack 3,
- Windows 2000.

Voir aussi le chapitre "Installation des composants Client", sous-chapitre "Informations préalables".

Communications

Pour permettre la communication entre les composants clients et les serveurs dans un environnement WINDOWS, le protocole de communication est TCP-IP Socket.

Chapitre 3. Installation de la partie Serveur

Installation

Le produit est livré sur support CD-Rom. Celui-ci contient les éléments suivants :

- une documentation succincte,
- Data.Cab (tout fichier du serveur au format compressé)
- setup.exe (exécutable d'installation),
- setup.ini (paramètres d'installation),
- VisualAge Pacbase 3.0.Msi (table d'installation pour Windows Installer).

L'installation requiert les droits Administrateur NT.

L'unité et le nom du répertoire peuvent être modifiés en cours d'installation. Par défaut, le serveur est installé dans le répertoire racine:

C:\Program Files\IBM\VisualAge Pacbase\

L'installation se déroule en trois étapes:

1. installation du Système, qui crée le répertoire \SYS\
2. installation de la base d'administration , qui crée le répertoire \CONFIG\ADMIN\ et le répertoire \DATA\ADMIN\
3. installation d'une base de développement, qui crée le répertoire \CONFIG\[nom_base]\ et le répertoire \DATA\[nom_base]\

Chaque étape peut être exécutée soit indépendamment, soit de manière enchaînée, mais toujours dans l'ordre indiqué.

Toutes les bases sont créées avec des données de tests d'installation et des paramètres de configuration.

Installation du Système.

L'exécution de setup.exe lance le programme d'installation qui guide l'utilisateur.

Après l'écran de BIENVENUE, certaines caractéristiques modifiables du poste s'affichent :

- Nom et Organisation du propriétaire
- Localisation d'installation (par défaut: C:\Program Files\IBM\VisualAge Pacbase\)

L'installation copie :

- le fichier de conversions, le fichier des messages produits par les procédures en cas d'erreur ou pour information
- les squelettes dans les différentes langues supportées par le produit
- les programmes
- les procédures
- l'utilitaire de création de base

Installation du Référentiel

Base Administration

Cette phase crée la base de l'Administrateur qui contrôle les bases utilisateurs, et ajoute des procédures 'batch' propres à l'espace administration. On ne peut installer qu'une seule base d'administration; elle se nomme ADMIN.

Trois informations sont demandées:

- La clé d'accès au produit (70 caractères)
- Le numéro de port de la base (situé entre 6000 et 9000)
- Le code langue de la base (par défaut, Anglais)

On peut modifier les répertoires proposés (par défaut: C:\Program Files\IBM\VisualAge Pacbase\).

Procédures exécutées : INAE, INGU, REST, VINS.

Note : L'installation de la base d'administration se fait, soit à la suite de celle du système, soit plus tard avec l'icône [CREER NOUVELLE BASE].

Base de Développement

Toute nouvelle Base de développement doit avoir été préalablement référencée dans la Base d'administration.

Le référencement d'une nouvelle base se fait avec Administrator workbench (voir les informations concernant son installation dans le chapitre Installation des composants Client, sous-chapitre Administrator & Developer workbench).

Note : La base de test BVAP livrée à l'installation est déjà déclarée dans la Base administration.

L'installation d'une nouvelle base crée les fichiers utiles au développeur pour gérer les entités dans le dictionnaire. On peut installer plusieurs bases de développement, chacune ayant son propre environnement.

Il faut renseigner 5 paramètres :

- Le nom de la base (8 car.)
- Le code de la base (4 car., tel que déclaré dans la base d'administration).
- Le numéro de port de la base (compris entre 6000 et 9000, unique pour chaque base).
- Le code langue de la base (par défaut, Anglais)
- Le code langue pour la génération (par défaut, Anglais)

On peut modifier les répertoires proposés (par défaut: C:\Program Files\IBM\VisualAge Pacbase\).

Procédures exécutées : REST, VINS

Créer nouvelle Base

Cette icône du menu "démarrer" installe la Base d'administration ou une base de développement, à tout moment après l'installation du système.

L'installation demande un code utilisateur de niveau d'autorisation 4. Dans la base d'administration de test, l'utilisateur de niveau 4 "ADMIN" est déjà déclaré.

Paramètres de Configuration

Pour chaque base , on crée un répertoire de Configuration avec

- 1- BvpServer.ini, qui contient les paramètres du serveur conversationnel.
- 2- Fhredir.cfg , qui contient les paramètres du "Rollback".

Liste des dates des composants

Procédure INSL

Cette procédure édite :

- la liste des programmes batch, programmes TP et leurs caractéristiques de génération.
- Cette liste doit être éditée, en cas de détection d'anomalies dans le fonctionnement du système, pour transmettre au support du produit les références de l'installation.

INSL - Script d'exécution

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      - LIST OF INSTALLED PROGRAMS AND FILES -
REM *
REM * -----

```

```

REM *
REM *
REM * INPUT:
REM * - ONE LINE : BVP PREFIX OF THE PRODUCT
REM *   (COL.3 LENGTH 3 )
REM *
REM * -----
REM *

Dim fso, dir, subdir, file, PathFile
Dim ListExclude
Dim regEx, retVal

Set WshShell = Wscript.CreateObject("Wscript.Shell")

' Set Path of programs :
' -----
Reg_PROD = "HKLM\SOFTWARE\IBM\BVP VisualAge Pacbase\"
Rep_INST = WshShell.RegRead (Reg_PROD)

Rep_PGM = Rep_INST & "Server\Sys\pgm"
Rep_CFG = Rep_INST & "Server\Config"

' Set List of cobol programs :
' -----
ListCobol = Rep_CFG & "\" & "ListCob.txt"
Set fso = WScript.CreateObject("Scripting.FileSystemObject")
Set dir = fso.GetFolder(Rep_PGM)
Set StreamLst = fso.CreateTextFile(ListCobol, true, false)
Set regEx = New RegExp

ListExclude = " BVPSEVER.EXE BVPDIAL.EXE BVPWSH.OCX "
ListExclude = " BVPLOCK.DLL BVPPAUSE.DLL " & ListExclude
ListExclude = " BVPSPLIT.EXE BVPTRANS.EXE " & ListExclude
ListExclude = " CODEPAGECONV.DLL VAPUTIL.DLL " & ListExclude
ListExclude = " BVP.ICO BVPSEVER.ICO " & ListExclude

For Each file In dir.Files
    regEx.Pattern = file.Name
    regEx.IgnoreCase = True
    retVal = regEx.Test(ListExclude)
    'Cobol program
    If retVal = False Then
        LineLst = Rep_PGM & "\" & file.Name & vbCRLF
        StreamLst.WriteLine(LineLst)
    End If
Next
Set StreamLst = Nothing

' Launch INSL :
' -----
' Debug Mode
' CREATE_LOG = True
CREATE_LOG = False

```

```

Set WshEnv = WshShell.Environment("PROCESS")
WshEnv("LIST") = ListCobol

INSL_LOG=chr(34) & Rep_CFG & "\BVPINSL.log" & chr(34)
INSL_OUT=chr(34) & Rep_CFG & "\BVPINSL.txt" & chr(34)
sCmd = Rep_PGM & "\Bvupdate.exe " & INSL_OUT
If ( CREATE_LOG = True ) Then
    sCmd = "cmd.exe /c " & sCmd & " 1>" & INSL_LOG & " 2>&1"
    Return = WshShell.Run( sCmd, 7 , TRUE)
Else
    Return = WshShell.Run( sCmd, 10 , TRUE)
End If

' End of job :
'-----
If ( Return <> 0 ) Then
    Display = "Abnormal end of Job."
else
    ' delete list of cobol programs
    Set FileD = fso.GetFile(ListCobol)
    FileD.Delete
    Display = "End of Job." & vbCRLF & vbCRLF & _
    "File " & INSL_OUT & " has been created."
End If

If ( CREATE_LOG = True ) Then
    Display = Display & vbCRLF & vbCRLF & _
    "File " & INSL_LOG & " has been created."
End If

WScript.Echo(Display)
Wscript.Quit (0)

```

Chapitre 4. Installation/Ré-installation des composants Client

Informations préalables

- Pour installer les composants Client de VisualAge Pacbase sur un poste Windows, vous devez avoir le profil Administrateur sur ce poste.
- L'installation des composants Client s'exécute avec InstallShield for Windows Installer (ISWi). Si Windows Installer n'est pas installé sur le poste, il sera automatiquement installé.
- Vous avez également besoin de Microsoft Windows Script, version 5.1 minimum. Vous pouvez la télécharger à partir de l'URL suivante :
www.microsoft.com/msdownload/vbscript/scripting.asp
- Les composants Administrator & Developer workbench et Outils eBusiness requièrent, pour leur fonction Aide en Ligne, le logiciel Netscape ou le logiciel Internet Explorer version 5.5.
- L'installation d'un composant Client n'exige pas l'installation préalable au niveau Serveur de la Base ou des Bases VA Pac à laquelle/auxquelles il se connectera.

Toutefois, le code de chaque Base VA Pac que vous allez fournir lors de l'installation d'un composant Client, devra être strictement réutilisé lors de l'installation de chacune de ces Bases dans la partie serveur.

- Par défaut, le répertoire racine de tous les composants Client VisualAge Pacbase est :

C:\Program Files\IBM\VisualAge Pacbase\

NOTE IMPORTANTE :

Les répertoires de cette racine ne sont modifiables qu'une seule fois, au début, c'est-à-dire à la première installation du premier composant.

Que cette racine ait été modifiée ou non, les autres composants s'installeront nécessairement sous cette même racine.

Par contre, lors de l'installation d'une version ultérieure d'un composant, l'installateur vous proposera :

- soit un rafraîchissement sous cette racine,
- soit une nouvelle racine, non modifiable, construite à partir de la précédente et dont le nom du dernier répertoire sera numériquement incrémenté.

Par exemple, si vous n'avez pas modifié la racine par défaut lors de l'installation initiale, la racine de la première ré-installation sera alors :
C:\Program Files\IBM\VisualAge Pacbase_1\

De même, la racine de la seconde ré-installation sera :
C:\Program Files\IBM\VisualAge Pacbase_2\

Principes et architectures de la communication

PREAMBULE : Les lignes qui suivent vous présentent les principes de communication entre les composants Client et le serveur VisualAge Pacbase.

Ces lignes contiennent des informations indispensables au choix du protocole de communication et au paramétrage du middleware associé. Ces informations vous seront également utiles lors d'installations ultérieures (autres composants Client ou nouvelles versions de composants déjà installés).

LES PROTOCOLES DE COMMUNICATION DISPONIBLES

- Si le serveur VisualAge Pacbase est installé sous Windows ou UNIX, le protocole VAP Socket devra être utilisé.
- Si le serveur est installé sous MVS/CICS, vous aurez le choix entre MVS CICS Socket et MVS CICS CPI-C, en fonction de l'installation effectuée sur le serveur (voir le chapitre correspondant).
- Si le serveur est installé sous MVS/IMS, vous aurez le choix entre MVS IMS Connect et MVS IMS CPI-C, en fonction de l'installation effectuée sur le serveur (voir le chapitre correspondant).

NOTE :

LE MIDDLEWARE

Le middleware doit toujours être installé sur chaque poste Développeur. Cette installation est automatiquement déclenchée à la première installation sous une racine donnée d'un des composants Client suivants : Administrator & Developer workbench, Station de Travail VisualAge Pacbase, les Outils eBusiness et Pacbase Web Connection. Pour ce dernier composant, cette installation est automatiquement déclenchée si l'option Serveur de Contexte est retenue.

Le middleware installé sur chaque poste Développeur assure ainsi la communication directe entre le ou les composants Client et le serveur.

Cependant, vous pouvez aussi opter pour une communication de type gateway.

Une gateway effectue une gestion centralisée et optimisée des accès serveur.

Dans ce contexte, le middleware est également à installer sur un serveur intermédiaire, en sélectionnant l'item Middleware dans la liste des composants Client installables (voir le sous-chapitre correspondant).

Les composants Client communiquent alors avec une passerelle (la Gateway VisualAge Pacbase) qui s'exécute sur ce serveur intermédiaire.

Cette option n'est pas disponible avec le composant Pacbase Web Connection.

LES FICHIERS DE LA COMMUNICATION

Pour les deux composants Administrator & Developer workbench et Station de Travail VisualAge Pacbase, le paramétrage de la communication s'articule nécessairement sur deux fichiers, bases.ini et vaplocat.ini.

Le fichier vaplocat.ini est également exploité par le composant Outils eBusiness.

Ces fichiers sont créés de façon transparente et se trouvent dans un répertoire nommé 'common'.

Une réinstallation n'a pas d'impact sur les fichiers bases.ini et vaplocat.ini. Un fichier base_new.ini est créé pour information uniquement; il contient la dernière version livrée de ce fichier.

IMPORTANT : Pour ajouter/supprimer ultérieurement des Bases VisualAge Pacbase, voire modifier certains paramètres liés à la communication, vous devrez modifier ces fichiers.

Les détails concernant la structure des données dans ces deux fichiers sont fournis à la fin de ce chapitre (Mise à jour des paramètres de la communication).

Le fichier bases.ini :

Le fichier bases.ini se trouve sur le poste de chaque développeur, dans le sous-répertoire 'common' du répertoire racine de l'installation.

Ce fichier établit la liste des Bases VisualAge Pacbase accessibles. A chaque Base est associé un nom de localisation (location en anglais).

Plusieurs Bases peuvent avoir une localisation identique. Ces localisations sont définies dans l'autre fichier (vaplocat.ini).

Le fichier vaplocat.ini :

- Lorsque la communication est directe, le fichier vaplocat.ini exploité se trouve sur chaque poste Développeur.
- Lorsque la communication est de type gateway, le fichier vaplocat.ini exploité se trouve sur le serveur intermédiaire.

Dans les deux cas, ce fichier est placé dans le sous-répertoire 'common' du répertoire racine de l'installation.

C'est dans ce fichier que sont décrite(s) la (ou les) localisation(s).

Une localisation :

- Identifie le protocole utilisé pour accéder au serveur VisualAge Pacbase,
- Donne les coordonnées physiques de ce serveur pour ce protocole,
- Définit les paramètres de communication nécessaires au bon fonctionnement du protocole.

Lancement de l'installation

Insérer dans le lecteur le CD-ROM livré.

L'exécution de setup.exe lance l'interface graphique du Wizard InstallShield qui va vous guider tout au long de l'installation.

Le premier panneau affiche le texte de la licence du runtime Java. Vous acceptez les termes de la licence ; le panneau suivant vous demande alors votre identification (Nom et Organisation).

NOTE : Tous les composants Client de VisualAge Pacbase sont installés en mode utilisation partagée pour le poste.

Ensuite, la liste des composants Client de l'offre VisualAge Pacbase s'affiche sur l'écran.

Choisissez le composant Client que vous souhaitez installer.

La suite de l'installation est décrite dans les sous-chapitres spécifiques à chaque composant.

Administrator & Developer workbench

La racine utilisée pour l'installation à effectuer dépend du contexte de cette installation. Pour plus de détails, reportez-vous à la première page de ce chapitre.

Dans le panneau suivant, sélectionnez le ou les éléments à installer :

- Administrator workbench

IMPORTANT :

Son installation est OBLIGATOIRE sur au moins un poste de votre site puisqu'elle vous permettra notamment de créer la ou les Bases VA Pac de votre site, des Bibliothèques, des Profils Utilisateur, et des utilisateurs.

- Developer workbench

Developer workbench inclut les modules suivants :

- le module Batch,
- le module eBusiness et trois outils eBusiness :
 - Générateur de Proxy
 - Editeur de Localisations (Location Editor)
 - Outil de test (Services Test Facility)

NOTE : L'installation de ces outils se fait dans un second temps par rapport au reste du module eBusiness.

- le Pont Rational Rose.

Dans le panneau suivant, vous indiquez le mode de communication utilisé (communication directe ou via gateway).

NOTE : Des informations IMPORTANTES concernant la communication vous sont données en début de chapitre.

Ce panneau ne s'affiche pas si vous avez déjà installé, sous la même racine, Administrator & Developer workbench ou la Station de Travail VisualAge Pacbase.

- Si vous choisissez la communication directe, le script d'installation du middleware s'enclenchera automatiquement, une fois l'installation du workbench terminée, et vous demandera de configurer les paramètres de la communication.

Pour la documentation de cette partie de l'installation, voir le sous-chapitre Middleware.

- Si vous choisissez gateway, renseignez ici l'adresse IP de cette gateway. L'installation du middleware sur le poste Développeur -- également nécessaire dans ce contexte -- sera donc automatiquement enclenchée après celle du workbench.

NOTE : Vous devrez donc aussi effectuer l'installation du middleware et la configuration de la communication au niveau du serveur intermédiaire supportant la gateway.

Ensuite, dans ce même contexte initial, vous indiquez la (première) Base à laquelle accèdera Administrator & Developer workbench.

Pour cela, un panneau vous propose de renseigner :

1. Le nom d'une Base VA Pac, déjà installée ou non au niveau serveur.

Les noms que vous saisissez ici seront ceux affichés dans l'assistant de connexion, indiquant ainsi à l'utilisateur la (ou les) Base(s) qu'il pourra sélectionner.

Le nom donné à chaque Base doit donc être suffisamment explicite pour qu'elles soient facilement identifiables dans la liste des Bases proposées par l'assistant de connexion.

2. Le code logique de cette Base.

Longueur maximum : 4 caractères.

Si cette Base n'est pas encore installée au niveau Serveur, notez bien le code que vous saisissez ; il devra être réutilisé lors de cette installation.

Les codes que vous saisissez ici seront également affichés dans l'assistant de connexion.

NOTE : La Base dédiée à l'administration du site est automatiquement créée. Son code logique est le code réservé : "****".

3. Le nom de la Localisation.

Longueur maximum : 20 caractères. Défaut : Location-1

Plusieurs Bases peuvent pointer sur la même localisation.

4. Pour finir, vous précisez le type de contrôle que le workbench appliquera à la connexion. Référez-vous à l'Administrateur de la Base concernée, pour qu'il y ait adéquation entre le ou les contrôles imposés au niveau serveur et leur implémentation au niveau du workbench.

Cet indicateur a trois valeurs possibles, à sélectionner via deux cases à cocher :

VAPac : indique que l'utilisateur aura à donner son code et son mot de passe pour se connecter à la Base VisualAge Pacbase.

Middleware : indique que l'utilisateur aura à donner son code et son mot de passe pour se connecter au système hôte (dans les deux zones affichées sous "Références Middleware" de l'assistant de connexion).

Si seule la case Middleware est cochée, le contrôle de la connexion à la Base VA Pac est assuré par le système de sécurité.

Si vous cochez les deux cases, l'utilisateur devra donner son code et son mot de passe pour la connexion au système hôte et pour la connexion à la Base.

Vous avez ainsi défini l'accès à une première Base VA Pac. Le script d'installation vous propose alors de définir la communication et la connexion avec une autre Base, et cela, autant de fois que nécessaire.

L'installation proprement dite peut ensuite démarrer; appuyez sur le bouton [INSTALL].

NOTE : Cette installation est suivie de celle des outils eBusiness (s'ils n'ont pas déjà été installés sous la même racine), et -- dans le contexte initial, défini ci-dessus -- de celle du middleware.

LES FICHIERS DE LANCEMENT

Les fichiers de lancement des modules sont les suivants :

```
admin.bat
batch.bat
eBusiness.bat
runcfm.bat
```

Ces fichiers se trouvent dans le répertoire racine de Administrator & Developer workbench (ADWorkbench).

MENU DEMARRER / CHOIX PROGRAMMES

A l'issue de cette installation, vous trouverez au niveau du bureau Windows, dans le menu Démarrer/choix Programmes, la section VisualAge Pacbase Components avec les sous-sections suivantes :

```
Administrator-Developer workbench
    Administration
    Batch
    eBusiness
```

Outils eBusiness

Les Outils eBusiness sont les suivants :

- Générateur de Proxy
- Editeur de Localisations (Location Editor)
- Outil de test (Services Test Facility)
- VisualAge Pacbase Connector

Cette installation permet l'utilisation des Outils eBusiness indépendamment de Developer workbench, sans communication avec un serveur VisualAge Pacbase. Les Outils eBusiness sont installés en tant que "features" et "tools" de VisualAge Java.

VisualAge Pacbase Connector a la particularité de ne s'exécuter que dans VisualAge Java.

La racine utilisée pour l'installation à effectuer dépend du contexte de cette installation. Pour plus de détails, reportez-vous à la première page de ce chapitre.

Pour démarrer l'installation, appuyez sur le bouton [INSTALL].

Le composant Middleware est automatiquement installé à la suite de l'installation des Outils eBusiness s'il n'est pas déjà présent sous la racine de l'installation courante. Vous aurez alors à renseigner certains paramètres de communication.

Pour la documentation de cette partie de l'installation, voir le sous-chapitre Middleware.

Le composant des Outils eBusiness peut fonctionner en mode communication par gateway. Dans ce cas, vous devrez installer le composant Middleware et configurer la communication au niveau du serveur intermédiaire supportant la gateway VisualAge Pacbase.

NOTE : Des informations IMPORTANTES concernant la communication vous sont données en début de chapitre.

Le middleware installé dans ce contexte particulier assure la communication entre les serveurs et les proxies générés.

Pour information, les paramètres de ce type de communication seront à positionner par chaque développeur en utilisant l'outil Location Editor inclus dans cette installation.

LES FICHIERS DE LANCEMENT

Les fichiers de lancement s'appellent :

- Pour le Générateur de Proxy :
vapGen.exe
- Pour l'Editeur de Localisation (Location Editor) :
vapLocationEditor.exe
- Pour l'outil de test (Services Test Facility) :
vapServicesTestFacility.exe

Ces fichiers se trouvent dans le répertoire racine des outils eBusiness (EBusinessTools).

NOTE : VisualAge Pacbase Connector s'exécute uniquement en tant qu'outil dans le cadre de VisualAge Java.

MENU DEMARRER / CHOIX PROGRAMMES

A l'issue de cette installation, vous trouverez au niveau du bureau Windows, dans le menu Démarrer/choix Programmes, la section VisualAge Pacbase Components avec les sous-sections suivantes :

eBusiness Tools
 Location Editor
 Proxy Generator
 Services Test Facility

Station de Travail VisualAge Pacbase

La racine utilisée pour l'installation à effectuer dépend du contexte de cette installation. Pour plus de détails, reportez-vous à la première page de ce chapitre.

Le premier panneau vous invite à sélectionner la langue dans laquelle l'interface de la Station de Travail VisualAge Pacbase apparaîtra.

La langue installée par défaut est l'anglais.

Le panneau suivant vous invite à sélectionner la méthode à installer.

NOTE : Si vous souhaitez installer une autre méthode sur le poste, le processus de cette installation sera à reprendre à partir du début.

Si elle est proposée, l'option "Local Install" est à sélectionner systématiquement.

NOTE : L'option "sub-features" est identique à l'option "feature" seule.

Dans le panneau suivant, vous sélectionnez :

- Le ou les modules à installer :
 - Pacdesign,
 - Pacbench.
- Le mode de connexion :
 - en mode "libre" : Choix par l'utilisateur de travailler en local ou connecté au Référentiel VisualAge Pacbase.
 - en mode "forcé" : Connexion systématique au Référentiel.

Dans le panneau suivant, vous indiquez le mode de communication utilisé (communication directe ou via gateway).

NOTE : Des informations IMPORTANTES concernant la communication vous sont données en début de chapitre.

Ce panneau ne s'affiche pas si vous avez déjà installé, sous la même racine, la Station de Travail VisualAge Pacbase ou Administrator & Developer workbench.

- Si vous choisissez la communication directe, le script d'installation du middleware s'enclenchera automatiquement, une fois l'installation de la Station de Travail terminée, et vous demandera de configurer les paramètres de la communication.

Pour la documentation de cette partie de l'installation, voir le sous-chapitre Middleware.

- Si vous choisissez gateway, renseignez ici l'adresse IP de cette gateway. L'installation du middleware sur le poste Développeur -- également nécessaire dans ce contexte -- sera automatiquement enclenchée après celle de la Station.

NOTE : Vous devrez donc aussi effectuer l'installation du middleware et la configuration de la communication au niveau du serveur intermédiaire supportant la gateway.

Ensuite, dans ce même contexte initial, vous indiquez la (première) Base à laquelle accèdera la Station de Travail VisualAge Pacbase.

Pour celà, un panneau vous propose de renseigner :

1. Le nom d'une Base VA Pac, déjà installée ou non au niveau serveur.

Les noms que vous saisissez ici seront ceux affichés dans la boîte de connexion, indiquant ainsi à l'utilisateur la (ou les) Base(s) qu'il pourra sélectionner.

Le nom donné à chaque Base doit donc être suffisamment explicite pour qu'elles soient facilement identifiables dans la liste des Bases proposées via la boîte de connexion.

NOTE : Si vous utilisez un fichier personnalisé des paramètres, indiquez, après le nom de la Base, le nom de ce fichier, encadré des signes "<" et ">".

Pour plus de détails sur ces paramètres, reportez-vous à la fin du sous-chapitre.

2. Le code logique de cette Base.

Longueur maximum : 4 caractères.

Si cette Base n'est pas encore installée au niveau Serveur, notez bien le code que vous saisissez ; il devra être réutilisé lors de cette installation.

Les codes que vous saisissez ici seront également affichés dans la boîte de connexion.

La Base dédiée à l'administration du site est automatiquement créée. Son code logique est le code réservé : "****".

NOTE : Un code logique doit être unique pour une Localisation donnée (voir le point 3. suivant).

3. Le nom de la Localisation.
Longueur maximum : 20 caractères.
Défaut : Location-1
Plusieurs Bases peuvent pointer sur la même localisation.
4. Pour finir, vous précisez le type de contrôle que la Station appliquera à la connexion. Référez-vous à l'Administrateur de la Base concernée, pour qu'il y ait adéquation entre le ou les contrôles imposés au niveau serveur et leur implémentation au niveau Station.
Cet indicateur a trois valeurs possibles, à sélectionner via deux cases à cocher :
VAPac : indique que l'utilisateur aura à donner son code et son mot de passe pour se connecter à la Base VisualAge Pacbase.
Middleware : indique que l'utilisateur aura à donner son code et son mot de passe, dans la boîte Identification Middleware, pour se connecter au système hôte. Si seule la case Middleware est cochée, le contrôle de la connexion à la Base VA Pac est assuré par le système de sécurité.
Si vous cochez les deux cases, l'utilisateur devra donner son code et son mot de passe pour la connexion au système hôte et pour la connexion à la Base.

Vous avez ainsi défini l'accès à une première Base VA Pac. Le script d'installation vous propose alors de définir la communication et la connexion avec une autre Base, et cela, autant de fois que nécessaire.

L'installation proprement dite peut ensuite démarrer; appuyez sur le bouton [INSTALL].

NOTE : Cette installation -- dans le contexte initial, défini ci-dessus -- est automatiquement suivie de celle du middleware.

LE FICHER DE LANCEMENT

Le fichier de lancement est le suivant :
pexec.exe

Ce fichier se trouve dans le répertoire racine de la Station de Travail VisualAge Pacbase (SPAC).

MENU DEMARRER / CHOIX PROGRAMMES

A l'issue de cette installation, vous trouverez au niveau du bureau Windows, dans le menu Démarrer/choix Programmes, la section VisualAge Pacbase Components avec les sous-sections suivantes :

```
WorkStation
    WorkStation
    WorkStation News
    <méthode> News
```

LE FICHER DES PARAMETRES DE L'INSTALLATION

Certains paramètres de l'installation de la Station de Travail sont positionnés dans le fichier Pacbase.dat.

La procédure d'installation de la Station de Travail crée automatiquement ce fichier dans le répertoire \SPAC\NNNL où "NNN" désigne le numéro de version et "L" le code langue de la version installée.

Le fichier Pacbase.dat - nécessairement conforme à la dernière installation effectuée - est donc utilisé par défaut lors du lancement de la Station.

Cependant, vous pouvez créer un ou plusieurs fichiers de paramétrage. Ceci peut s'avérer utile dans le cas assez rare où plusieurs méthodes sont installées sur un poste. Le changement de méthode sera ainsi facilité à la reconnexion.

Les noms de ces fichiers sont libres mais doivent respecter la norme des fichiers DOS. Il est recommandé de leur donner l'extension .dat.

Vous devez constituer ces fichiers à l'image du fichier Pacbase.dat et les placer dans le même répertoire que lui.

En cas de réinstallation de la Station, les fichiers *.dat que vous avez créés ne seront pas supprimés.

DESCRIPTION DU FICHER DES PARAMETRES

Chacune des lignes qui composent ce fichier possède la structure suivante :

- un identifiant numérique sur trois caractères en positions 1 à 3;
- le libellé de la ligne, dont la position est libre;
- la valeur du paramètre, entre crochets ([et]), dont la position est également libre.

Voici un exemple de fichier PACBASE.DAT :

```
001 Version Station           [300F]
002 Serveur                   [PACBASE]
003 Gestionnaire de communication [MWCOM]
004 Paramètres de communication [MWCOM]
```


005	Système d'exploitation	[WINDOWS]
006	Méthode	[MER]
007	Disque EXE	[C]
008	Disque EXE (défaut)	[C]
009	Disque Données système	[C]
010	Disque Données utilisateur	[C]
011	Mode d'exécution connexion	[E]

Le fichier Pacbase.dat ne doit pas être détruit.

Les valeurs possibles pour le paramètre Méthode sont :

VALEUR DU PARAMETRE	NOM DE LA METHODE
MER	MERISE
DON	YSM
FAA	IFW
ADM	SSADM (en anglais seulement)
OMT	OMT

ATTENTION : Les paramètres 001 à 005 et 011 ne sont pas modifiables.

Pacbase Web Connection

INSTALLATION PREALABLE REQUISE

Pacbase Web Connection requiert l'installation d'un interpréteur PERL (version 5.0 minimum) que vous pouvez télécharger à partir de l'url suivante : <http://www.perl.com>

Vous installerez cet interpréteur selon les indications fournies ci-dessous.

INSTALLATION PROPREMENT DITE

La racine utilisée pour l'installation à effectuer dépend du contexte de cette installation. Pour plus de détails, reportez-vous à la première page de ce chapitre.

Vous choisissez ensuite entre installer le serveur de contexte ou Pacbase Web Generator ou les deux.

En effet, vous pouvez installer le serveur de contexte et le générateur sur deux postes différents ou sur le même poste développeur à partir duquel seront générées les pages html.

Par ailleurs, le poste où s'exécutera le serveur HTTP doit impérativement disposer de l'interpréteur PERL pour exploiter le script qui établira la connexion entre le serveur HTTP et le serveur de contexte.

NOTE : Le serveur HTTP et le serveur de contexte peuvent être installés sur des postes différents.

L'installateur vous demande de saisir le nom du répertoire où est installé l'interpréteur PERL.

Si PERL n'est pas encore installé, l'installateur crée le répertoire ad-hoc, nommé Pacwebperl. Un sous-répertoire BIN contient nécessairement les fichiers cgicgi.pl et cgi-lib.pl.

L'installation proprement dite peut alors démarrer. Pour cela, appuyer sur le bouton [Install].

L'installation du Serveur de Contexte est automatiquement suivie de celle du composant Middleware, sauf s'il est déjà installé sous la racine de l'installation courante. Lors de l'installation du composant Middleware, vous renseignez certains paramètres de communication.

Pour la documentation de cette partie de l'installation, voir le sous-chapitre Middleware.

NOTE : Des informations IMPORTANTES concernant la communication vous sont données en début de chapitre.

LE FICHER DE LANCEMENT

Le fichier de lancement est :

Pacweb.exe

Ce fichier se trouve dans le répertoire racine de Pacbase Web Connection (PacWeb).

Installation du serveur de contexte comme service NT

Sous Windows/NT, le serveur de contexte peut être installé en tant que service NT. Dans ce cas, le lancement du serveur de contexte se fait automatiquement ou par l'intermédiaire de la boîte de dialogue commune à tous les services.

L'installation du serveur de contexte comme service se fait en entrant la commande :

```
pacweb -i [<N° de socket>]
```

depuis le répertoire d'installation du serveur de contexte. Le numéro de socket est optionnel. Sa valeur par défaut est 2345.

Lors de l'installation du service, le serveur de contexte n'est pas lancé, son lancement se fait grâce au bouton " démarrer " de la boîte de dialogue des services.

Lors du redémarrage de la machine, il est automatiquement lancé.

La désinstallation du service se fait par la commande :

```
pacweb -d [<N° de socket>]
```

Ces commandes d'installation et de désinstallation de service ne peuvent s'exécuter correctement que par un utilisateur possédant le droit d'ouvrir une session en tant que service.

Middleware

L'installation spécifique du composant Middleware, sur une machine dédiée (serveur intermédiaire), n'est nécessaire que lorsqu'une communication par gateway est mise en oeuvre.

En effet, le composant Middleware est automatiquement installé, juste après la première installation (sous une racine donnée) d'un des autres composants Client.

NOTE : Pour le composant Pacbase Web Connection, c'est l'installation du sous-composant Serveur de Contexte qui déclenche l'installation du Middleware.

La racine utilisée pour l'installation à effectuer dépend du contexte de cette installation. Pour plus de détails, reportez-vous à la première page de ce chapitre.

Pour utiliser Administrator & Developer workbench ou la Station de Travail VA Pac, les paramètres de localisation de vos Bases VisualAge Pacbase doivent toujours être renseignés.

NOTE : Des informations IMPORTANTES concernant la communication vous sont données en début de chapitre.

- Si la communication est assurée via la gateway VisualAge Pacbase, l'installation du Middleware sur ce serveur intermédiaire vous demande de définir la localisation nécessaire pour une première Base VA Pac.

NOTE : Vous saisissez le nom de la localisation dans la zone 'New location name'.

ATTENTION :

Si vous avez plus d'une localisation à définir, que ce soit pour la même Base ou pour gérer plusieurs Bases, vous définissez ces localisations supplémentaires directement dans le fichier vaplocat.ini.

Pour les détails concernant la mise à jour de ce fichier, reportez-vous au sous-chapitre Informations complémentaires, section Mise à jour des paramètres de la communication.

- Si la communication est directe, les localisations vous sont automatiquement proposées ; en effet, elles ont été prédéfinies, juste auparavant, lors de l'installation de Administrator & Developer workbench ou de la Station de Travail.

Ensuite, et quel que soit le composant Client concerné, vous aurez à saisir un certain nombre de paramètres différents, selon le protocole utilisé.

- Si la communication est assurée via la gateway VisualAge Pacbase, ces paramètres vous seront demandés lors de l'installation du Middleware sur ce serveur intermédiaire.
- Si la communication est directe, ces paramètres vous sont demandés lors de l'installation automatique du middleware.

LISTE DES PARAMETRES

- VAP SOCKET

IP address : Adresse IP et port utilisé par le serveur VA Pac

- MVS CICS SOCKET

IP address : Adresse IP et port utilisé par le serveur VA Pac

Transaction code : Code de la transaction CICS du Moniteur de Communication VisualAge Pacbase.

Code Page : Valeur identifiant le codage des caractères utilisé par le serveur VisualAge Pacbase. La valeur 1147 correspond à l'EBCDIC français

- MVS CICS/IMS CPI-C

Destination-id entry : BVPSCPI (valeur par défaut). Si vous modifiez cette valeur, elle doit être identique à celle positionnée dans Symbolic destination name, paramètre inclus dans la configuration de ce protocole de communication.

Code page : Valeur identifiant le codage des caractères utilisé par le serveur VisualAge Pacbase. La valeur 1147 correspond à l'EBCDIC français

- MVS IMS Connect

IP address : Adresse IP et port utilisé par le serveur VA Pac

Transaction code : Code de la transaction IMS du Moniteur de Communication VisualAge Pacbase.

Code Page : Valeur identifiant le codage des caractères utilisé par le serveur VisualAge Pacbase. La valeur 1147 correspond à l'EBCDIC français

Data Store : Nom du lien vers IMS défini dans IMS Connect (IMS Data Store ID)

RACF group : Nom du groupe RACF pour IMS Connect.

L'installation proprement dite peut maintenant démarrer. Pour celà, appuyez sur le bouton [Install].

Mise à jour des paramètres de la communication

Le fichier bases.ini

Vous aurez besoin de mettre à jour le fichier bases.ini pour rajouter ou supprimer une Base, ou pour modifier des paramètres de communication.

Par défaut, le chemin d'accès à ce fichier est :

C:\Program Files\IBM\VisualAge Pacbase\Common\

NOTE : Tous les paramètres pouvant être présents dans le fichier bases.ini ne sont pas explicités ici. En effet, certains de ces paramètres permettent des réglages fins concernant le fonctionnement du middleware, notamment utilisés lors de l'exploitation des proxies (générées par les Outils eBusiness). Dans ce contexte, ces paramètres sont exploités séparément du fichier bases.ini et sont documentés dans le Manuel de Référence Interface de Programmation des Proxies.

Le format de ce fichier répond aux normes standard des fichiers .ini de Windows.

Chaque section de ce fichier définit une configuration permettant d'accéder à une Base VisualAge Pacbase. Le nom de cette section est placé entre crochets [Nom de la section].

Le nom de chaque section sera présenté à l'utilisateur lors de sa connexion. Dans la liste ainsi affichée, il choisira sa base de connexion. C'est pourquoi ce nom doit être le plus explicite possible. D'autant plus que vous pouvez définir plusieurs configurations/sections pour une Base, qui doivent donc être facilement identifiables; vous définissez ainsi plusieurs options de communication pour une même Base.

NOTE : Avec la Station de Travail VA Pac, vous pouvez utiliser un fichier personnalisé des paramètres. Dans ce cas, indiquez, après le nom de la Base, le nom de ce fichier, encadré des signes "<" et ">". Pour plus de détails sur ces paramètres, reportez-vous plus haut, à la fin du sous-chapitre "Station de Travail VisualAge Pacbase".

CONTENU DE CHAQUE SECTION

Les paramètres de chaque section sont listés ci-dessous à raison d'un paramètre par ligne.

- baseCode = code logique de la Base VisualAge Pacbase (obligatoire)
Longueur maximum : 4 caractères

NOTE : Dans le cas de la Station de Travail, ce code logique doit être unique dans le fichier bases.ini pour une localisation donnée.

- signOn = paramètre précisant le mode de contrôle de l'utilisateur à la connexion. Ce paramètre obligatoire prend une des trois valeurs suivantes :
 - VAPac : indique que l'utilisateur n'aura à donner son code et son mot de passe que pour sa connexion à la Base VisualAge Pacbase.
 - Middleware : indique que l'utilisateur n'aura à donner son code et son mot de passe que pour sa connexion au système hôte. Le contrôle de la connexion à la Base étant assuré par RACF (ou équivalent).
 - VAPac Middleware : indique que l'utilisateur devra donner son code et son mot de passe pour la connexion au système hôte et pour la connexion à la Base (option par défaut).
- communicationAdapter = ce paramètre indique le mode de communication retenu.
 - DIRECT : middleware local
 - GATEWAY : middleware distant (via gateway VisualAge Pacbase)

Les paramètres suivants varient suivant l'option choisie.

PARAMETRES POUR ADAPTATEUR DIRECT (MIDDLEWARE LOCAL)

- locationsFile = ce paramètre indique le chemin et le nom du fichier qui contient les définitions des localisations. défaut: ..\common\vaplocat.ini
ATTENTION : Il est très fortement déconseillé de modifier la valeur par défaut de ce paramètre.
- location = nom de la localisation pour la Base
Longueur maximum : 20 caractères.
Défaut : Location-1
Plusieurs Bases peuvent pointer sur la même localisation.
RAPPEL : Une localisation identifie le protocole de communication utilisé pour accéder au serveur VisualAge Pacbase et les coordonnées physiques de ce serveur pour ce protocole.
- traceFile = chemin et nom du fichier qui réceptionnera la trace d'exécution du middleware.

Par défaut, ce fichier est automatiquement créé (avec timestamp) dans le sous-répertoire VapTrace.

- `traceLevel` = niveau de détail de la trace d'exécution du middleware. Les valeurs possibles sont :
 - 0 : pas de trace
 - 1 : trace des erreurs (défaut)
 - 2 : trace standard non détaillée
 - 3 : trace pour informations
 - 4 et + : trace de debug
- `codePageFile` = chemin et nom du fichier contenant les tables de conversion des codes page.
Défaut: ..\middleware\CharConv.txt

PARAMETRES POUR ADAPTATEUR GATEWAY (MIDDLEWARE DISTANT)

- `host` = nom ou adresse IP du host où la gateway VisualAge Pacbase est installée.
Défaut : 127.0.0.1 pour host local
- `port` = valeur du port IP où la gateway réceptionne les requêtes clientes.
Défaut : 5647
- `location` = nom de la localisation pour la Base
Longueur maximum : 20 caractères.
Défaut : Location-1
Plusieurs Bases peuvent pointer sur la même localisation.
RAPPEL : Une localisation identifie le protocole de communication utilisé pour accéder au serveur VisualAge Pacbase et les coordonnées physiques de ce serveur pour ce protocole.

Le fichier vaplocat.ini

Vous aurez besoin de mettre à jour le fichier vaplocat.ini pour rajouter ou supprimer une Base, ou pour éventuellement modifier d'autres paramètres décrits ci-dessous.

Par défaut, le chemin d'accès à ce fichier est :
C:\Program Files\IBM\VisualAge Pacbase\Common\

NOTE : Tous les paramètres du fichier vaplocat.ini ne sont pas explicités ici. En effet, certains de ces paramètres permettent des réglages fins concernant le fonctionnement du middleware, notamment utilisés lors de l'exploitation des proxies (générées par les Outils eBusiness). Dans ce contexte particulier, ces paramètres sont édités via l'outil Editeur de Localisations (Location Editor) et sont donc documentés via son aide en ligne.

Pour ajouter une Base VisualAge Pacbase, créez une ligne où vous indiquez le nom de la localisation entre les caractères "<" et ">".

La longueur maximum de ce nom est de 20 caractères.

Selon le protocole sélectionné, vous aurez à saisir des paramètres différents (un par ligne) :

- VAP SOCKET

```
<NomLocalisation>
COMM_TYPE=SOCKET
MONITOR=BVPSCPI
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
IXO_ADDRESS=127.0.0.1 3000
```
- MVS CICS SOCKET

```
<NomLocalisation>
COMM_TYPE=TCPMVS
MONITOR=BVPSSOC
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=1147
IXO_ADDRESS=127.0.0.1 3000
IXO_TRANSID=V303
```
- MVS CICS/IMS CPI-C

```
<NomLocalisation>
COMM_TYPE=CPIC
MONITOR=BVPSCPI
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=1147
```
- IMS Connect

```
<NomLocalisation>
COMM_TYPE=TCPIMS
MONITOR=BVPSCPI
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=1147
IXO_ADDRESS=127.0.0.1 3000
IXO_TRANSID=P300CPI
IXO_DATASTORE=IMSC
IXO_RACFGROUP=FR42
```

EXPLICATION DES PARAMETRES :

La liste ci-dessous est triée par ordre alphabétique des paramètres.

- COMM_TYPE :

Ce paramètre désigne le protocole de communication utilisé.

Les valeurs possibles sont :

SOCKET : Serveur VA Pac sous Windows ou UNIX, utilisant TCP/IP.

TCPMVS : Serveur VA Pac sous MVS/CICS ou MVS/IMS, utilisant un listener TCP/IP.

CPIC : Serveur VA Pac sous MVS/CICS, utilisant le protocole CPI-C.

TCPIMS : Serveur VA Pac sous MVS/IMS, utilisant le protocole IMS Connect.

- IXO_ADDRESS :

Adresse IP et port utilisé par le serveur VA Pac. Le numéro de port doit correspondre au numéro de port indiqué lors de la configuration du serveur.

- IXO_DATASTORE :

Nom du lien vers IMS défini dans IMS Connect (IMS DataStore ID).

- IXO_RACFGROUP :

Nom du groupe RACF pour IMS Connect.

- IXO_TIMEOUT :

Temps maximum pendant lequel une station de travail attend la réponse du serveur avant de signaler une erreur de communication.

L'unité de ce paramètre est la seconde. Sa valeur par défaut est 30.

- IXO_TRANSID :

Code de la transaction CICS

- HOST_ENCODING :

Identifie le codage des caractères utilisé par le serveur VisualAge Pacbase.

La valeur 1147 correspond à l'EBCDIC français.

- MESSAGE_LENGTH :

La valeur de ce paramètre est obligatoirement 31744.

- MONITOR :

Code du moniteur de communication pour VisualAge Pacbase, c'est-à-dire BVPSCPI, ou BVPSSOC pour MVS CICS SOCKET.

NOTE :

Désinstallation de composants Client

Pour désinstaller un composant Client, utilisez le service Windows NT "Ajout/Suppression de Programmes" dans le Panneau de configuration.

Vous pouvez également reprendre le CD d'installation et activer la fonction "Remove".

Chapitre 5. Tests

Liste des utilitaires principaux

Le tableau résume l'ensemble des utilitaires de gestion de la Base d'administration et de la Base de développement.

Script	Description
ARCH	Archivage du journal de la Base administration
PACS	Sauvegarde de la Base administration (SAVE)
REOR	Réorganisation de la Base administration
REST	Restauration de la Base administration
ARCH	Archivage du journal de la Base développement
PACS	Sauvegarde de la Base développement (SAVE)
PACS	Gestion des bibliothèques de la Base développement (MLIB)
REOR	Réorganisation de la Base développement
REST	Restauration de la Base développement
UPDT	Mise à jour batch de la Base développement
GPRT	Edition génération
PACX	Extraction de bibliothèques (EXLI)
PACX	Extraction du journal (EXPJ)
PACX	Extraction d'entités (EXTR)
PACS	Extraction de sous-réseaux (UXSR)

Tests de l'installation

Ces tests comprennent plusieurs phases :

- . Tests de mises à jour TP et batch, d'édition-génération
- . Tests des procédures d'administration
- . Tests des utilitaires d'extraction.

Tests de mise à jour TP/batch , Edition-Génération

Ces tests comprennent les étapes suivantes :

- Tests d'utilisation en TP :
 - Démarrer le serveur de la base de test.
 - Dérouler les enchaînements d'écrans.
 - Effectuer des mises à jour.
- Tests de mise à jour batch :
 - Exécuter la procédure UPDT.
- Tests d'édition et de génération de programmes :
 - Exécuter la procédure GPRT.

Tests des procédures de la base Administration

Ces tests comprennent les étapes suivantes, à exécuter dans l'ordre :

- Archivage du journal créé lors des tests d'utilisation : exécuter la procédure ARCH donnant un fichier PJ(1).
- Sauvegarde de la Base administration : exécuter la procédure PACS (option SAVE) donnant un fichier PC(1).
- Réorganisation de la sauvegarde séquentielle PC(1) de la Base administration : exécuter la procédure REOR donnant un fichier PC(2).
- Restauration de la Base administration à partir de l'archive PJ(1) et de la sauvegarde de la base PC(2) : exécuter la procédure REST.

Pour tous ces tests, le serveur de la Base d'administration doit être arrêté.

Tests des procédures de la base de Développement

Ces tests comprennent les étapes suivantes, à exécuter dans l'ordre :

- Archivage du journal créé lors des tests d'utilisation : exécuter la procédure ARCH donnant un fichier PJ(1).
- Sauvegarde directe de la Base de développement : exécuter la procédure PACS (option SAVE) donnant un fichier PC(1).
- Gestionnaire de bibliothèque : ajout/suppression de bibliothèque dans la Base de développement : exécuter la procédure PACS(mlib) donnant un fichier PC(2).
- Réorganisation de la sauvegarde séquentielle PC(2) de la Base de développement : exécuter la procédure REOR donnant un fichier PC(3).
- Restauration de la Base de développement à partir de l'archive PJ(1) et de la sauvegarde de la base PC(3) : exécuter la procédure REST.

Pour tous ces tests, les fichiers de la Base de développement doivent être fermés dans le conversationnel.

Après restauration et réouverture des fichiers de la Base de développement, il est recommandé de refaire quelques tests rapides en mode conversationnel afin de s'assurer du bon fonctionnement de l'application.

Tests des utilitaires d'extraction

Ces tests ont pour objet d'exécuter les procédures d'extraction de la base.

Ils comprennent les étapes suivantes, à exécuter dans l'ordre :

- Extraction d'une bibliothèque du réseau sous forme de mouvements : exécuter la procédure PACX option EXLI.
- Extraction d'entités d'une bibliothèque : exécuter la procédure PACX option EXTR.
- Extraction de mouvements sélectionnés et/ou listes de mouvements sélectionnés à partir du journal archivé (PJ) : exécuter la procédure PACX option EXPJ.

Pour tous ces tests, les fichiers de la base peuvent être ouverts dans le conversationnel.

Chacun des jobs peut enchaîner sur une mise à jour UPDT permettant de vérifier la validité des mouvements extraits.

Chapitre 6. Réinstallation Serveur

Une réinstallation de la partie système du Serveur VisualAge Pacbase est à effectuer à la suite de la réception d'une sous-version apportant des corrections d'anomalies ou des développements ponctuels sur la version installée.

Le service Windows NT -Ajout/Suppression de programmes- permet d'intervenir avec les options "Mise à jour/Suppression".

Réinstallation

- La mise à jour de la partie système est proposée automatiquement, si vous avez une version antérieure déjà installée.

Désinstallation

La désinstallation de la partie système se fait par le service "ajout/suppression de programmes" du panneau de configuration. (application 'VisualAge Pacbase Server')

On propose, au même moment, de supprimer de la base de Registre les accès aux données mais de sauvegarder ces données sous un autre répertoire \DATA.

Suppression des données

La suppression d'une base de données est une opération définitive. Il est conseillé de faire une sauvegarde avant.

Les différentes étapes sont des commandes manuelles :

- Supprimer le répertoire des données de la base et tous les sous-répertoires dépendants.
- Supprimer le répertoire:

Winnt\Profiles\AllUsers\Start Menu\Programs\VisualAge
Pacbase Server\[nom_base]

- Supprimer du registre HK_Local_Machine la clé:
Software\IBM\BVP VisualAge Pacbase\Server\BVP_DATA\[nom_base]

Chapitre 7. Reprises

Reprises des bases des versions 2.0 et 2.5

Opérations à effectuer

L'installation de la version 3.0 nécessite, d'une part, la reprise des fichiers AG (commandes édition-génération), AE AP (paramètres utilisateurs) et AB AC (fichiers PEI) dans la nouvelle Base administration, et d'autre part la reprise de l'ancienne Base de développement.

Enchaînement à effectuer :

Il se déroule en six étapes :

1) Sauvegarde de tous les anciens fichiers nécessaires par exécution des procédures suivantes dans l'ancienne version.

- SAVE : sauvegarde de la Base de développement (PC),
- PARM : sauvegarde des paramètres utilisateur (PE),
- SVAG : sauvegarde des commandes d'édition-génération (PG),
- SVPE : sauvegarde de l'environnement PEI (PP).

2) Mise en place de la Base administration 3.0.

Cette étape nécessite le déroulement de l'installation de la Base administration.

Elle permet de créer les fichiers AN, AR, AY, AJ et GU.

Elle consiste en l'exécution des procédures suivantes :

- création de la Base administration proprement dite,
 - INGU : création et initialisation du fichier GU des codes utilisateurs,
 - REST : initialisation de la Base administration avec les données d'installation (valoriser la clé d'accès),
 - VINS : installation du Modèle administration (voir la description en annexe en fin de manuel),
- réorganisation éventuelle de la Base administration si un message de fin d'installation le demande :
 - PACS : sauvegarde de la Base administration (save),
 - REOR : réorganisation de la Base administration,
 - ARCH : initialisation du fichier journal de la Base administration,

- REST : restauration de la Base administration,
- reprise des données de l'ancienne base,
- PE25 : reprise des paramètres utilisateurs à partir du fichier PE généré lors de l'étape 1,
- SAVE : sauvegarde de la Base administration.

3) Reprise d'une base de développement.

Cette étape nécessite l'installation préalable de la base VA Pacbase correspondante avec les données de test.

Puis consiste en l'exécution des procédures suivantes :

- PC25 : reprise et réorganisation de l'ancienne Base de développement à partir de la sauvegarde de l'ancienne base créée dans l'étape 1,
- REOR : réorganisation de la nouvelle Base de développement,
- REST : restauration de la nouvelle Base de développement à partir de la sauvegarde obtenue précédemment,
- VINS : installation du Modèle développement de la nouvelle base (voir la description en annexe en fin de manuel),

L'exécution des trois procédures suivantes est facultative et peut être demandée par souci d'optimisation.

- SAVE : sauvegarde de la nouvelle Base de développement,
- REOR : réorganisation de la nouvelle Base de développement,
- REST : restauration de la Base de développement à partir du fichier de sauvegarde obtenu en sortie de la procédure de réorganisation précédente.

Les étapes 4 5 et 6 sont facultatives.

4) Reprise des commandes d'édition-génération.

Elle consiste en l'exécution des procédures suivantes :

- PG20 : Reprise d'un fichier PG 2.0,
- PG25 : Reprise d'un fichier PG 2.5

5) Reprise des paramètres de Pac/Transfer (UV).

Elle consiste en l'exécution de la procédure suivante :

- UV25 : reprise des données du fichier UV.

6) Reprise des fichiers PEL.

Elle consiste en l'exécution de la procédure suivante :

- PP25 : reprise des données du fichier PP.

Reprise des paramètres utilisateur (PE25)

PE25 - Présentation générale

Principe

Cette procédure (PE25) permet de reprendre le fichier PE, issu de la sauvegarde des paramètres utilisateurs par la procédure PARM, pour les intégrer dans la Base administration.

Condition d'exécution

Les fichiers de la Base administration doivent être fermés dans le transactionnel.

Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

Résultat obtenu

La Base administration intégrant les paramètres utilisateurs issus de la version 2.0 ou 2.5.

PE25 - Entrées / Traitements / Résultats

Une ligne * avec Code utilisateur et Mot de passe.

Une ligne 'A' permettant de préciser le code et le nom de l'administrateur.

- Si le code ou le nom de l'administrateur n'est pas renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'A'	Code ligne
3	8	bbbbbbbb	Code de l'administrateur
11	36		Nom de l'administrateur

Une ligne 'B' par base permettant de préciser les caractéristiques des Bases de développement qui seront gérées dans la nouvelle Base administration soit :

- le code de la base : c'est le code logique qui sera indiqué lors de la restauration de la base.

- le nom de la base
- Si le code ou le nom de la base n'est pas renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'B'	Code ligne
3	4	bbbb	Code base logique
7	36		Nom de la base

PE25 - Description des étapes

Traitement des paramètres utilisateur (PE) : PTU920

Code	Nom physique	Type	Libellé
PAC7EN	Rép. save : PE25	Entrée	Paramètres utilisateur ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements paramètres des utilisateurs (longueur=310)
PAC7ET	Rép. user : PE25ET920	Etat	Compte-rendu en cas d'anomalie

Mise en forme des mouvements : PAF900

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PAC7AN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur

Code	Nom physique	Type	Libellé
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7GY	Rép. tmp : WGY	Entrée	Mouvements de mise à jour
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements mis en forme (doit pouvoir contenir tous les mouvements en entrée plus les mouvements d'annulation élémentaire générés par les mouvements d'annulation multiple) (longueur = 170)
PAC7ME	Rép. tmp : WME	Sortie	Fichier de travail (longueur=372)
PAC7MW	Rép. tmp : WMW	Sortie	Fichier de travail (longueur=170)
PAC7MX	Rép. tmp : WMX	Sortie	Fichier de travail (longueur=743)
PAC7MY	Rép. tmp : WMY	Sortie	Fichier de travail (longueur=743)

Mise à jour de la Base Administration : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Sortie	Données de la Base administration
PAC7AN	Base Admin - Rép. base : AN	Sortie	Index de la Base administration
PAC7AY	Base Admin - Rép. base : AY	Sortie	Extension de la Base administration
PAC7AJ	Base Admin - Rép. Journal : AJ	Sortie	Journal de la Base administration
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base administration

Code	Nom physique	Type	Libellé
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	nul	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	nul	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		Etat	Récapitulatif des mouvements erronés (longueur=132)

La liste des mouvements propres à un utilisateur est précédée d'une bannière portant son code.

Codes retour :

- 0 : OK sans erreur
- 2 : erreur warning
- 4 : erreur grave

PE25 - Script d'exécution

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      RETRIEVAL OF PE FILE
REM *
REM * -----
REM *
<job id=PE25>

<script language="VBScript">
Dim MyProc
MyProc = "PE25"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

```

```

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1029" ))
'-----
Call StateList (base, statusL)

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTU920"))
'-----
WshEnv("PAC7MB") = Fic_Input
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU920","PAC7EN",Rep_SAVE & "\PE.old")
Call BvpEnv("PTU920","PAC7ET",Rep_USR & "\PE25ET920.txt")
Call BvpEnv("PTU920","PAC7GY",Rep_TMP & "\WGY.tmp")
Return = WshShell.Run("BVPТУ920.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU920")

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVPAF900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PE25IEA15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PE25IFA15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")

```

```

Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP )

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Reprise de la base de Développement (PC25)

PC25 - Présentation générale

Principe

La procédure (PC25) permet de reprendre le fichier PC issu de la sauvegarde de l'ancienne Base de développement en un fichier PC nouveau format.

Condition d'exécution

aucune.

Edition obtenue

Cette procédure édite un compte-rendu donnant le nombre de manuels transformés en rapports, les warning possibles sur les entités utilisateur, appels de formats guides et description d'états (données extension), le code de la nouvelle Base de développement et le nombre d'enregistrements en sortie du fichier PC.

Résultat obtenu

Le résultat de cette procédure est une image séquentielle de la base de développement au nouveau format. Ce nouveau fichier PC doit servir d'entrée à l'étape obligatoire suivante : la réorganisation.

PC25 - Remarques sur les données reprises

Eclatement du descriptif commentaire (-G)

Le descriptif commentaire est éclaté en plusieurs descriptifs.

- Commentaires proprement dits
Ils regroupent les commentaires et les alias COBOL (-GC).

Attention :

Si en version 2.0 ou 2.5, le type de la ligne de documentation n'est pas adapté à l'entité (ex : ligne de génération sur une rubrique) il deviendra un commentaire.

- Lignes de génération

Elles regroupent les types G, P, V et Z (-GG).

- Paramètres de génération

On y retrouve le type de ligne O (-GO).

- Gestion des libellés d'erreur

Ils regroupent les types C, D, F ,S ,T , U (-GE).

- Appels d'entités par relations

On y retrouve le type de ligne R (-CR).

- Cas particulier du format guide

Le type présent sur le descriptif du format guide détermine la valeur du type de la fiche de définition, soit 'C' pour les commentaires, 'G' pour les compléments de génération ou 'O' pour les options de générations. Les appels de formats guides pourront donc être visualisables en -GC, -GG ou -GO.

Attention :

Si en version 2.0 ou 2.5 sur un même descriptif, le type a plusieurs valeurs, un message d'erreur est édité : une intervention manuelle est demandée pour corriger l'incohérence.

Là encore si l'appel de format guide est 'Généré' ou 'Option de dialogue' à tort, l'appel devient un commentaire.

Remarque importante :

Si en version 2.n , une ligne de type autre que commentaire est écrasée par une ligne de commentaire, du fait de l'éclatement des -G, cet écrasement ne pourra se retrouver en version 3.0, il faudra intervenir manuellement dans la nouvelle base.

Type de table des structures de données

Les structures de données de type table (G, T, M, N) et vue logique (V) conservent leur type. Tous les autres types (fichiers...) deviennent Z. L'état n'est plus supporté par la structure de donnée, le type J disparaît.

Transformation des manuels U

les manuels sont transformés en rapports, leurs codes sont complétés par 'EIBM'.

Données extension : entités utilisateurs, formats guides, maquettes des états.

Les enregistrements suite de ces entités disparaissent, il n'y a plus un index par enregistrement principal et un par enregistrement suite. Des données extension sont créées qui concatènent les informations présentes sur les anciens enregistrements. Ces données longues peuvent avoir une longueur de 1000 et sont réparties sur plusieurs enregistrements. Un seul index est désormais créé qui pointe sur le premier de ces enregistrements.

Remarque importante :

Si en version 2.n, un enregistrement suite est modifié dans une sous-bibliothèque, la reprise ne peut répercuter cette modification sur la donnée extension au niveau des bibliothèques inférieures de cette sous-bibliothèque.

Ainsi, si la reprise détecte une mise à jour dans une ou des sous-bibliothèques d'un même enregistrement suite, les warning suivants sont édités :

- "Risque d'incohérence de la fiche de l'entité utilisateur \$xx xxxxxx dans la bibliothèque xxx".
- "Risque d'incohérence de la ligne xxx de documentation sous format guide xxxxxxxx dans la bibliothèque xxx".
- "Risque d'incohérence des libellés de la ligne xxx de l'état xxx dans la bibliothèque xxx"

L'utilisateur devra intervenir manuellement si ces enregistrements ne sont pas en phase avec ceux de la version 2.0 ou 2.5.

PC25 - Entrées / Traitements / Résultats

Une ligne * avec le code de la nouvelle Base de développement.

Cette ligne est facultative si le code base attribué en version 2.5 convient. Ce code base doit avoir été défini dans la Base administration.

Si aucun code base n'est renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'**'	Code ligne
3	4	bbbb	Code nouvelle base

PC25 - Description des étapes

Traitements généraux : PTU911

Code	Nom physique	Type	Libellé
PAC7MC	Rép. save : PC25	Entrée	Image séquentielle du réseau ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur
PAC7PB	Rép. tmp : WPC	Sortie	Premier enregistrement des données (longueur=153)
PAC7PE	Rép. tmp : WPE	Sortie	Fiches des O.E.U. (version 2.5), maquettes d'états et des commentaires (excepté les appels de formats guides) (longueur=193)
PAC7PG	Rép. tmp : WPG	Sortie	Descriptions des formats guides et des commentaires avec appels de formats guides (longueur=193)
PAC7PL	Rép. tmp : WPL	Sortie	Fiches des rapports, de leurs descriptions et des manuels (longueur=193)
PAC7PZ	Rép. tmp : WPZ	Sortie	Entités utilisateurs et descriptifs de leurs occurrences (version 2.5.) (longueur=193)
PAC7PF	Rép. tmp : WPF	Sortie	Enregistrements autres (longueur=153)
PAC7PM	Rép. tmp : WPM	Sortie	Fichier compte-rendu (longueur=48)
PAC7ET	Rép. user : PC25ET911	Etat	Compte-rendu en cas d'absence du code base

Traitement des manuels et des rapports : PTU909

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7PB	Rép. tmp : WPC	Entrée	Premier enregistrement des données
PAC7PL	Rép. tmp : WPL	Entrée	Fiches des rapports, de leurs descriptions et des manuels

Code	Nom physique	Type	Libellé
PAC7PI	Rép. tmp : WPI	Sortie	Fiches des rapports et de leurs descriptions triés et reformatés (longueur=153)
PAC7PM	Rép. tmp : WPM	Entrée/Sortie	Fichier compte-rendu

Traitement des commentaires PTU92A

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7PG	Rép. tmp : WPG	Entrée	Descriptions des formats guides et de leurs appels dans les commentaires
PAC7PM	Rép. tmp : WPM	Entrée/Sortie	Compte-rendu
PAC7PE	Rép. tmp : WPH	Sortie	Descriptions des formats guides et de leurs appels dans les commentaires (longueur=193)

Traitement des méta-entités : PTU912

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7PZ	Rép. tmp : WPZ	Entrée	Entités utilisateurs (version 2.5)
PAC7PB	Rép. tmp : WPC	Entrée	Premier enregistrement des données
PAC7ZP	Rép. tmp : WZP	Sortie	Enregistrements du Modèle développement (fiches et descriptions) (longueur=193)

Traitement des maquettes d'états : PTU919

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7PE	Rép. tmp : WPE	Entrée	Fiches des O.E.U (version 2.5), des maquettes d'état et des commentaires (excepté les appels de formats guides)
PAC7PB	Rép. tmp : WPC	Entrée	Premier enregistrement des données

Code	Nom physique	Type	Libellé
PAC7PH	Rép. tmp : WPH	Entrée	Descriptions des formats-guides et de leurs appels dans les commentaires
PAC7PM	Rép. tmp : WPM	Entrée/Sortie	Fichier compte-rendu
PAC7ZP	Rép. tmp : WEP	Sortie	Fiches des O.E.U (version 2.5), des maquettes d'états, et des commentaires (avec appels de formats guides) (longueur=193)
PAC7PO	Rép. tmp : WPO	Sortie	Commentaires (excepté les appels de formats guides) (longueur=153)
PAC7PD	Rép. tmp : WPD	Sortie	Premier enregistrement des données (longueur=153)

Traitement des entités utilisateurs : PTU913

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7PX	Rép. tmp : WEP	Entrée	Fiches des O.E.U (version 2.5), des maquettes d'états et des commentaires (avec appels de formats guides)
PAC7PZ	Rép. tmp : WZP	Entrée	Modèle de développement (fiches et descriptifs) et descriptions des O.E.U (version 2.5)
PAC7PB	Rép. tmp : WPD	Entrée	Premier enregistrement des données
PAC7ZP	Rép. tmp : WZX	Sortie	Données extension du modèle développement et des entités utilisateurs, des maquettes d'états et des commentaires (avec appels de formats guides) (longueur=193)
PAC7PD	Rép. tmp : WPB	Sortie	Premier enregistrement des données (longueur=153)

Tri des données longues : PTU91A

Code	Nom physique	Type	Libellé
PAC7PZ	Rép. tmp : WZX	Entrée	Données extension intermédiaires
PAC7ZP	Rép. tmp : WXZ	Sortie	Données longues triées (longueur=193)

Fusion des fichiers : PTU914

Cette étape reconstitue l'image séquentielle finale à partir des fichiers intermédiaires issus des étapes précédentes.

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Fichier des libellés d'erreur
PAC7ZP	Rép. tmp : WXZ	Entrée	Données extension triées
PAC7PO	Rép. tmp : WPO	Entrée	Commentaires (sans appel de formats guides)
PAC7PD	Rép. tmp : WPD	Entrée	Premier enregistrement des données
PAC7PI	Rép. tmp : WPI	Entrée	Fiches des rapports et de leurs descriptions
PAC7PF	Rép. tmp : WPF	Entrée	Enregistrements autres
PAC7PM	Rép. tmp : WPM	Entrée	Fichier compte-rendu
PAC7PC	Rép. save : PC (+1)	Sortie	Image séquentielle du réseau (version 3.0)
PAC7ET	Rép. user : PC25ET914	Etat	Compte-rendu de la reprise

PC25 - Script d'exécution

```
REM * -----  
REM *      VISUALAGE PACBASE  
REM *  
REM * -----  
REM *      RETRIEVAL OF PC FILE  
REM *  
REM * -----  
REM *  
<job id=PC25>  
  
<script language="VBScript">  
MyProc = "PC25"  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>
```

```

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

If base = "ADMIN" Then
Call Msg_Log (Array("1028",base))
Wscript.Quit (0)
End If

Call Msg_Log (Array("1022" , "PTU911"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU911","PAC7MC",Rep_SAVE & "\PC25")
Call BvpEnv("PTU911","PAC7PB",Rep_TMP & "\WPC.tmp")
Call BvpEnv("PTU911","PAC7PE",Rep_TMP & "\WPE.tmp")
Call BvpEnv("PTU911","PAC7PF",Rep_TMP & "\WPF.tmp")
Call BvpEnv("PTU911","PAC7PG",Rep_TMP & "\WPG.tmp")
Call BvpEnv("PTU911","PAC7PL",Rep_TMP & "\WPL.tmp")
Call BvpEnv("PTU911","PAC7PZ",Rep_TMP & "\WPZ.tmp")
Call BvpEnv("PTU911","PAC7PM",Rep_TMP & "\WPM.tmp")
Call BvpEnv("PTU911","PAC7ET",Rep_USR & "\PC25ET911.txt")
Return = WshShell.Run("BVPTU911.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU911")

Call Msg_Log (Array("1022" , "PTU909"))
'-----
Call BvpEnv("PTU909","PAC7PI",Rep_TMP & "\WPI.tmp")
Call BvpEnv("PTU909","PAC7PM",Rep_TMP & "\WPM.tmp")
Return = WshShell.Run("BVPTU909.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU909")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\WPL.tmp")

Call Msg_Log (Array("1022" , "PTU92A"))
'-----
Call BvpEnv("PTU92A","PAC7PE",Rep_TMP & "\WPH.tmp")
Call BvpEnv("PTU92A","PAC7PM",Rep_TMP & "\WPM.tmp")
Return = WshShell.Run("BVPTU92A.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU92A")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\WPG.tmp")

Call Msg_Log (Array("1022" , "PTU912"))
'-----
Call BvpEnv("PTU912","PAC7ZP",Rep_TMP & "\WZP.tmp")
Return = WshShell.Run("BVPTU912.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU912")

```

```

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WPZ.tmp")

Call Msg_Log (Array("1022" , "PTU919"))
'-----
Call BvpEnv("PTU919","PAC7PD",Rep_TMP & "\\WPD.tmp")
Call BvpEnv("PTU919","PAC7PE",Rep_TMP & "\\WPE.tmp")
Call BvpEnv("PTU919","PAC7PH",Rep_TMP & "\\WPH.tmp")
Call BvpEnv("PTU919","PAC7PO",Rep_TMP & "\\WPO.tmp")
Call BvpEnv("PTU919","PAC7ZP",Rep_TMP & "\\WEP.tmp")
Return = WshShell.Run("BVPTU919.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU919")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WPC.tmp")
Call DelFile (Rep_TMP & "\\WPE.tmp")
Call DelFile (Rep_TMP & "\\WPH.tmp")

Call Msg_Log (Array("1022" , "PTU913"))
'-----
Call BvpEnv("PTU913","PAC7PB",Rep_TMP & "\\WPD.tmp")
Call BvpEnv("PTU913","PAC7PD",Rep_TMP & "\\WPB.tmp")
Call BvpEnv("PTU913","PAC7PX",Rep_TMP & "\\WEP.tmp")
Call BvpEnv("PTU913","PAC7PZ",Rep_TMP & "\\WZP.tmp")
Call BvpEnv("PTU913","PAC7ZP",Rep_TMP & "\\WZX.tmp")
Return = WshShell.Run("BVPTU913.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU913")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WPD.tmp")
Call DelFile (Rep_TMP & "\\WEP.tmp")
Call DelFile (Rep_TMP & "\\WZP.tmp")

Call Msg_Log (Array("1022" , "PTU91A"))
'-----
Call BvpEnv("PTU91A","PAC7PZ",Rep_TMP & "\\WZX.tmp")
Call BvpEnv("PTU91A","PAC7ZP",Rep_TMP & "\\WXZ.tmp")
Return = WshShell.Run("BVPTU91A.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU91A")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WZX.tmp")

Call Msg_Log (Array("1022" , "PTU914"))
'-----
WshEnv("PAC7PC") = Rep_SAVE & "\\PC.new"
Call BvpEnv("PTU914","PAC7ET",Rep_USR & "\\PC25ET914.txt")
Return = WshShell.Run("BVPTU914.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU914")

```



```

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_SAVE & "\PC")

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Reprise commandes d'édition-génération (PG20)

PG20 - Présentation générale

Principe

La procédure (PG20) permet de reprendre le fichier PG image séquentielle des commandes d'édition-génération de la version 2.0 au nouveau format.

Elle met à jour la Base de développement avec les commandes d'édition-génération, et la Base administration avec les lignes de Script (présentes sur l'écran GP visualisé en option C4 version 2.0).

Condition d'exécution

Les fichiers de la Base Administration et de la Base de Développement doivent être fermés dans le transactionnel.

Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

Remarques

La mise à jour rejettera tout mouvement à insérer dans une bibliothèque ou une session inexistante dans la nouvelle base. Le fichier PG peut contenir des commandes spécifiques à une bibliothèque ou une session purgée ultérieurement.

La mise à jour d'une commande d'édition-génération pour une entité ne peut pas s'effectuer si l'entité n'existe pas. Exemple : pour la commande GCP PROGRA, le programme PROGRA devra être présent dans la base.

Les codes utilisateurs présents dans le fichier PG et absents dans la Base administration sont automatiquement créés pour les utilisateurs ayant des Scripts.

PG20 - Entrées / Traitements / Résultats

Une ligne * avec code utilisateur, mot de passe et code de la Base de développement concernée par les lignes de Script mises à jour dans la Base administration.

Si le code utilisateur ou le code de la base n'est pas renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	mot de passe
22	4	cccc	Code base

PG20 - Description des étapes

Formatage des commandes d'édition génération : PTU908

Code	Nom physique	Type	Libellé
PAC7IN	Rép. save : PG20	Entrée	Commandes édition-génération ancienne version
PAC7OU	Rép. save : OLDPG	Sortie	Commandes édition-génération reformatées (longueur=150)

Traitement des commandes d'édition génération : PTU921

Code	Nom physique	Type	Libellé
PAC7PG	Rép. save : OLDPG	Entrée	Commandes édition-génération ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements des commandes d'édition-génération (longueur=310)
PAC7GZ	Rép. tmp : WGZ	Sortie	Mouvements des lignes de Script (longueur=310)

Code	Nom physique	Type	Libellé
PAC7ET	Rép. user : PGnnET921	Etat	Compte-rendu d'anomalies (nn= 20 ou 25 , n° version)

Mise en forme des mouvements : PAF900

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7GY	Rép. tmp : WGZ	Entrée	Mouvements de mise à jour
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements mis en forme (doit pouvoir contenir tous les mouvements en entrée plus les mouvements d'annulation élémentaire générés par les mouvements d'annulation multiple) (longueur=170)
PAC7ME	Rép. tmp : WME	Sortie	Fichier de travail (longueur=372)
PAC7MW	Rép. tmp : WMW	Sortie	Fichier de travail (longueur=170)
PAC7MX	Rép. tmp : WMX	Sortie	Fichier de travail (longueur=743)
PAC7MY	Rép. tmp : WMY	Sortie	Fichier de travail (longueur=743)

Mise à jour de la Base de développement : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Sortie	Données de la Base de Développement
PAC7AN	Rép. base : AN	Sortie	Index de la Base de Développement

Code	Nom physique	Type	Libellé
PAC7AY	Rép. base : AY	Sortie	Extension de la Base de Développement
PAC7AJ	Rép. journal : AJ	Sortie	Journal de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7DC	Nul	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	nul	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	nul	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		Etat	Récapitulatif des mouvements erronés (longueur=132)

La liste des mouvements propres à un utilisateur est précédée d'une bannière portant son code.

Codes retour :

- 0 : OK sans erreur
- 2 : erreur warning
- 4 : erreur grave

Mise en forme des mouvements : PAF900

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PAC7AN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7GY	Rép. tmp : WGY	Entrée	Mouvements de mise à jour
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements mis en forme (doit pouvoir contenir tous les mouvements en entrée plus les mouvements d'annulation élémentaire générés par les mouvements d'annulation multiple) (longueur = 170)
PAC7ME	Rép. tmp : WME	Sortie	Fichier de travail (longueur=372)
PAC7MW	Rép. tmp : WMW	Sortie	Fichier de travail (longueur=170)
PAC7MX	Rép. tmp : WMX	Sortie	Fichier de travail (longueur=743)
PAC7MY	Rép. tmp : WMY	Sortie	Fichier de travail (longueur=743)

Mise à jour de la Base Administration : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Sortie	Données de la Base administration
PAC7AN	Base Admin - Rép. base : AN	Sortie	Index de la Base administration
PAC7AY	Base Admin - Rép. base : AY	Sortie	Extension de la Base administration
PAC7AJ	Base Admin - Rép. Journal : AJ	Sortie	Journal de la Base administration
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur

Code	Nom physique	Type	Libellé
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	nul	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	nul	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		Etat	Récapitulatif des mouvements erronés (longueur=132)

La liste des mouvements propres à un utilisateur est précédée d'une bannière portant son code.

Codes retour :

- 0 : OK sans erreur
- 2 : erreur warning
- 4 : erreur grave

PG20 - Script d'exécution

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      RETRIEVAL OF PG FILE SINCE 2.0
REM *
REM * -----
REM *
<job id=PG20>

<script language="VBScript">

```

```

Dim MyProc
MyProc = "PG20"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1029" ))
'-----
Call StateList (base, statusL)

If c_error = 1 then Wscript.Quit (1) End If

'-PG- Retrieval for the Aministration database and
'the Development database
'THUS the same steps are executed is this procedure :
' = PTUBAS + ... PACA15 ... x 2 : concurrent access on AR
'For ADMIN <Semlock> and <Semadmin>' are set
' to Administration database
'For XXXX (database typed) <Semlock> is set
' to Development database
' and <Semadmin> is set to Administration database

Call Msg_Log (Array("1022" , "PTU908"))
'-----
WshEnv("PAC7IN") = Rep_SAVE & "\PG20"
WshEnv("PAC7OU") = Rep_SAVE & "\OLDPG"
Return = WshShell.Run("BVPTU908.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU908")

Call Msg_Log (Array("1022" , "PTU921"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU921","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PTU921","PAC7GZ",Rep_TMP & "\WGZ.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU921","PAC7PG",Rep_SAVE & "\OLDPG")
Call BvpEnv("PTU921","PAC7ET",Rep_USR & "\PG20ET921.txt")
Return = WshShell.Run("BVPTU921.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU921")

'for the ADMIN base(<Z> space) :

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"

```

```

WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900", "PAC7GY", Rep_TMP & "\WGZ.tmp")
Call BvpEnv("PAF900", "PAC7ME", Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900", "PAC7MV", Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900", "PAC7MW", Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900", "PAC7MX", Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900", "PAC7MY", Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVPAF900.EXE" , 1, TRUE)
If Return <> 0 then
Call Msg_Log (Array("1025", _
                    "PAF900/ADMinistration", Return))
End If

If Return = 0 then

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("SEMLOCK") = Rep_ABASE & "\LO"
WshEnv("SEADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
WshEnv("PAC7IE") = Rep_AUSR & "\PG20IE1A15.txt"
WshEnv("PAC7IF") = Rep_AUSR & "\PG20IF1A15.txt"
Call BvpEnv("PACA15", "PAC7ME", Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15", "PAC7MV", Rep_TMP & "\WMV.tmp")
WshEnv("PAC7RB") = "NUL"
WshEnv("PAC7RY") = "NUL"
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
If Return <> 0 then
Call Msg_Log (Array("1025", _
                    "PACA15/ADMinistration", Return))
End If

End If

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\WGZ.tmp")
Call DelFile (Rep_TMP & "\WME.tmp")
Call DelFile (Rep_TMP & "\WMV.tmp")
Call DelFile (Rep_TMP & "\WMW.tmp")
Call DelFile (Rep_TMP & "\WMX.tmp")
Call DelFile (Rep_TMP & "\WMY.tmp")
Return = 0
' To continue

'For the DEV base(<Y> space) :

```



```

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVP900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PG20IE2A15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PG20IF2A15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7RB") = "NUL"
WshEnv("PAC7RY") = "NUL"
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Reprise commandes d'édition-génération (PG25)

PG25 - Présentation générale

Principe

La procédure (PG25) permet de reprendre le fichier PG image séquentielle des commandes d'édition-génération de la version 2.5 au nouveau format.

Elle met à jour la Base de développement avec les commandes d'édition-génération, et la Base administration avec les lignes de Script (présentes sur l'écran GP visualisé en option C4 version 2.5).

Condition d'exécution

Les fichiers de la Base administration et de la Base de développement doivent être fermés dans le transactionnel.

Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

Remarques

La mise à jour rejettera tout mouvement à insérer dans une bibliothèque ou une session inexistante dans la nouvelle base. Le fichier PG peut contenir des commandes spécifiques à une bibliothèque ou une session purgée ultérieurement.

La mise à jour d'une commande d'édition-génération pour une entité ne peut s'effectuer si l'entité n'existe pas. Exemple : pour la commande GCP PROGRA, le programme PROGRA devra être présent dans la base.

Les codes utilisateurs présents dans le fichier PG et absents dans la Base administration sont automatiquement créés pour les utilisateurs ayant des Scripts.

PG25 - Entrées / Traitements / Résultats

Une ligne * avec code utilisateur, mot de passe et code de la Base de développement concernée par les lignes de Script mises à jour dans la Base administration.

Si le code utilisateur ou le code de la base n'est pas renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	mot de passe
22	4	cccc	Code base

PG25 - Description des étapes

Traitement des commandes d'édition génération : PTU921

Code	Nom physique	Type	Libellé
PAC7PG	Rép. save : OLDPG	Entrée	Commandes édition-génération ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements des commandes d'édition-génération (longueur=310)
PAC7GZ	Rép. tmp : WGZ	Sortie	Mouvements des lignes de Script (longueur=310)
PAC7ET	Rép. user : PGnnET921	Etat	Compte-rendu d'anomalies (nn= 20 ou 25 , n° version)

Mise en forme des mouvements : PAF900

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7GY	Rép. tmp : WGZ	Entrée	Mouvements de mise à jour

Code	Nom physique	Type	Libellé
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements mis en forme (doit pouvoir contenir tous les mouvements en entrée plus les mouvements d'annulation élémentaire générés par les mouvements d'annulation multiple) (longueur=170)
PAC7ME	Rép. tmp : WME	Sortie	Fichier de travail (longueur=372)
PAC7MW	Rép. tmp : WMW	Sortie	Fichier de travail (longueur=170)
PAC7MX	Rép. tmp : WMX	Sortie	Fichier de travail (longueur=743)
PAC7MY	Rép. tmp : WMY	Sortie	Fichier de travail (longueur=743)

Mise à jour de la Base de développement : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Sortie	Données de la Base de Développement
PAC7AN	Rép. base : AN	Sortie	Index de la Base de Développement
PAC7AY	Rép. base : AY	Sortie	Extension de la Base de Développement
PAC7AJ	Rép. journal : AJ	Sortie	Journal de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7DC	Nul	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	nul	Sortie	Mouvements erronés UPDT (longueur=80)

Code	Nom physique	Type	Libellé
PAC7RY	nul	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		Etat	Récapitulatif des mouvements erronés (longueur=132)

La liste des mouvements propres à un utilisateur est précédée d'une bannière portant son code.

Codes retour :

- 0 : OK sans erreur
- 2 : erreur warning
- 4 : erreur grave

Mise en forme des mouvements : PAF900

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PAC7AN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7GY	Rép. tmp : WGY	Entrée	Mouvements de mise à jour
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements mis en forme (doit pouvoir contenir tous les mouvements en entrée plus les mouvements d'annulation élémentaire générés par les mouvements d'annulation multiple) (longueur = 170)
PAC7ME	Rép. tmp : WME	Sortie	Fichier de travail (longueur=372)

Code	Nom physique	Type	Libellé
PAC7MW	Rép. tmp : WMW	Sortie	Fichier de travail (longueur=170)
PAC7MX	Rép. tmp : WMX	Sortie	Fichier de travail (longueur=743)
PAC7MY	Rép. tmp : WMY	Sortie	Fichier de travail (longueur=743)

Mise à jour de la Base Administration : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Sortie	Données de la Base administration
PAC7AN	Base Admin - Rép. base : AN	Sortie	Index de la Base administration
PAC7AY	Base Admin - Rép. base : AY	Sortie	Extension de la Base administration
PAC7AJ	Base Admin - Rép.Journal : AJ	Sortie	Journal de la Base administration
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	nul	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	nul	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		Etat	Récapitulatif des mouvements erronés (longueur=132)

La liste des mouvements propres à un utilisateur est précédée d'une bannière portant son code.

Codes retour :

- 0 : OK sans erreur
- 2 : erreur warning
- 4 : erreur grave

PG25 - Script d'exécution

```
REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      RETRIEVAL OF PG FILE SINCE 2.5
REM *
REM * -----
<job id=PG25>

<script language="VBScript">
Dim MyProc
MyProc = "PG25"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1029" ))
'-----
Call StateList (base, statusL)

If c_error = 1 then Wscript.Quit (1) End If

'-PG- Retrieval for the Aministration database
'and the Development database
'THUS the same steps are executed is this procedure :
' = PTUBAS + ... PACA15 ... x 2 : concurrent access on AR
'For ADMIN <Semlock> and <Semadmin>' are set to
' Administration database
'For XXXX (database typed) <Semlock> is set to
'Development database
' and <Semadmin> is set to Administration database

Call Msg_Log (Array("1022" , "PTU921"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU921","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PTU921","PAC7GZ",Rep_TMP & "\WGZ.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU921","PAC7PG",Rep_SAVE & "\OLDPG")
```

```

Call BvpEnv("PTU921","PAC7ET",Rep_USR & "\PG25ET921.txt")
Return = WshShell.Run("BVPTU921.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU921")

'for the ADMIN base(<Z> space) :

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGZ.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVPAF900.EXE" , 1, TRUE)
If Return <> 0 then
Call Msg_Log (Array("1025","PAF900/ADMinistration", Return))
End If

If Return = 0 then
Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_ABASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PG25IE1A15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PG25IF1A15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
If Return <> 0 then
Call Msg_Log (Array("1025",_
"PACA15/ADMinistration", Return))
End If

End If

Call Msg_Log (Array("1024"))
'-----

```



```

Call DelFile (Rep_TMP & "\WGZ.tmp")
Call DelFile (Rep_TMP & "\WME.tmp")
Call DelFile (Rep_TMP & "\WMV.tmp")
Call DelFile (Rep_TMP & "\WMW.tmp")
Call DelFile (Rep_TMP & "\WMX.tmp")
Call DelFile (Rep_TMP & "\WMY.tmp")
Return = 0
' To continue

'For the DEV base(<Y> space) :

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVP900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PG25IE2A15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PG25IF2A15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7RB") = "NUL"
WshEnv("PAC7RY") = "NUL"
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

```

```
Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>
```

Reprise PEI (PP25)

PP25 - Présentation générale

Principe

Cette procédure permet de reprendre le fichier PP image séquentielle de l'environnement de production (PEI) de la version 2.0 ou 2.5 et met à jour la Base de développement 3.0.

Condition d'exécution

Les fichiers de la Base de développement doivent être fermés dans le transactionnel.

Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

Résultat obtenu

Cette procédure génère un fichier mouvement contenant les environnements de production existants, la liste des entités générées, les environnements par défaut (-GO de la Bibliothèque), la liste des sessions de production au format 3.0, et met à jour la Base de développement 3.0.

Remarques

Tout mouvement de mise à jour dans une Bibliothèque ou une session inexistante dans la nouvelle Base sera rejeté.

Le fichier PP peut contenir des environnements avec des codes bibliothèques ou des sessions non encore créées ou purgées ultérieurement dans la Base 2.0 ou 2.5.

PP25 - Entrées / Traitements / Résultats

Une ligne * avec Code utilisateur et Mot de passe

Si le code utilisateur n'est pas renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	mot de passe

PP25 - Description des étapes

Traitement de l'environnement de production : PTU923

Code	Nom physique	Type	Libellé
PAC7PP	Rép. save : OLDPP	Entrée	Sauvegarde de l'environnement de production ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements des environnements de production (longueur=310)
PAC7ET	Rép. user : PP25ET923	Etat	Compte-rendu en cas d'anomalie

Mise en forme des mouvements : PAF900

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7GY	Rép. tmp : WGZ	Entrée	Mouvements de mise à jour

Code	Nom physique	Type	Libellé
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements mis en forme (doit pouvoir contenir tous les mouvements en entrée plus les mouvements d'annulation élémentaire générés par les mouvements d'annulation multiple) (longueur=170)
PAC7ME	Rép. tmp : WME	Sortie	Fichier de travail (longueur=372)
PAC7MW	Rép. tmp : WMW	Sortie	Fichier de travail (longueur=170)
PAC7MX	Rép. tmp : WMX	Sortie	Fichier de travail (longueur=743)
PAC7MY	Rép. tmp : WMY	Sortie	Fichier de travail (longueur=743)

Mise à jour de la Base de développement : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Sortie	Données de la Base de Développement
PAC7AN	Rép. base : AN	Sortie	Index de la Base de Développement
PAC7AY	Rép. base : AY	Sortie	Extension de la Base de Développement
PAC7AJ	Rép. journal : AJ	Sortie	Journal de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7DC	Nul	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	nul	Sortie	Mouvements erronés UPDT (longueur=80)

Code	Nom physique	Type	Libellé
PAC7RY	nul	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		Etat	Récapitulatif des mouvements erronés (longueur=132)

La liste des mouvements propres à un utilisateur est précédée d'une bannière portant son code.

Codes retour :

- 0 : OK sans erreur
- 2 : erreur warning
- 4 : erreur grave

PP25 - Script d'exécution

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      RETRIEVAL OF PP FILE
REM *
REM * -----
REM *
<job id=PP25>

<script language="VBScript">
Dim MyProc
MyProc = "PP25"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTU923"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU923","PAC7GY",Rep_TMP & "\WGY.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU923","PAC7PP",Rep_SAVE & "\OLDPP")
Call BvpEnv("PTU923","PAC7ET",Rep_USR & "\PP25ET923.txt")
Return = WshShell.Run("BVPTU923.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU923")

```

```

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900", "PAC7GY", Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900", "PAC7ME", Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900", "PAC7MV", Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900", "PAC7MW", Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900", "PAC7MX", Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900", "PAC7MY", Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVPAF900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15", "PAC7IE", Rep_USR & "\PP25IE1A15.txt")
Call BvpEnv("PACA15", "PAC7IF", Rep_USR & "\PP25IF1A15.txt")
Call BvpEnv("PACA15", "PAC7ME", Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15", "PAC7MV", Rep_TMP & "\WMV.tmp")
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Reprise paramètres Pac/Transfer (UV25)

UV25 - Présentation générale

Principe

La procédure (UV25) permet de reprendre le fichier UV des paramètres de Pac/Transfer de la version 2.0 ou 2.5 au nouveau format.

Elle met à jour la Base administration.

Condition d'exécution

Les fichiers de la Base administration doivent être fermés dans le transactionnel.

Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

UV25 - Entrées / Traitements / Résultats

Une ligne * avec code utilisateur, mot de passe et code de la Base de développement concernée par les transferts.

Si le code utilisateur ou le code de la base n'est pas renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	mot de passe
22	4	cccc	Code base

UV25 - Description des étapes

Traitement des paramètres de transfert : PTU922

Code	Nom physique	Type	Libellé
PAC7UV	Rép. base : UV25	Entrée	Paramètres de transfert de l'ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur

Code	Nom physique	Type	Libellé
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements des paramètres de transfert (longueur=310)
PAC7ET	Rép. user : UV25ET922	Etat	Compte-rendu en cas d'anomalie

Mise en forme des mouvements : PAF900

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PAC7AN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7GY	Rép. tmp : WGY	Entrée	Mouvements de mise à jour
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements mis en forme (doit pouvoir contenir tous les mouvements en entrée plus les mouvements d'annulation élémentaire générés par les mouvements d'annulation multiple) (longueur = 170)
PAC7ME	Rép. tmp : WME	Sortie	Fichier de travail (longueur=372)
PAC7MW	Rép. tmp : WMW	Sortie	Fichier de travail (longueur=170)
PAC7MX	Rép. tmp : WMX	Sortie	Fichier de travail (longueur=743)
PAC7MY	Rép. tmp : WMY	Sortie	Fichier de travail (longueur=743)

Mise à jour de la Base Administration : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Sortie	Données de la Base administration

Code	Nom physique	Type	Libellé
PAC7AN	Base Admin - Rép. base : AN	Sortie	Index de la Base administration
PAC7AY	Base Admin - Rép. base : AY	Sortie	Extension de la Base administration
PAC7AJ	Base Admin - Rép.Journal : AJ	Sortie	Journal de la Base administration
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	nul	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	nul	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		Etat	Récapitulatif des mouvements erronés (longueur=132)

La liste des mouvements propres à un utilisateur est précédée d'une bannière portant son code.

Codes retour :

- 0 : OK sans erreur
- 2 : erreur warning
- 4 : erreur grave

UV25 - Script d'exécution

```
REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      RETRIEVAL OF UV FILE
REM *
REM * -----
REM *
<job id=UV25>

<script language="VBScript">
Dim MyProc
MyProc = "UV25"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTU922"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU922","PAC7ET",Rep_USR & "\UV25ET922.txt")
Call BvpEnv("PTU922","PAC7GY",Rep_TMP & "\WGY.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU922","PAC7UV",Rep_BASE & "\UV25")
Return = WshShell.Run("BVPTU922.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU922")

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVPAF900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
```

```

WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15", "PAC7IE", Rep_USR & "\UV25IEA15.txt")
Call BvpEnv("PACA15", "PAC7IF", Rep_USR & "\UV25IFA15.txt")
Call BvpEnv("PACA15", "PAC7ME", Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15", "PAC7MV", Rep_TMP & "\WMV.tmp")
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Reprise mouvements MB (MB25)

MB25 - Présentation générale

Principe

Cette procédure permet de reprendre des mouvements MB de la version 2.0 ou 2.5 vers la version 3.0.

Condition d'exécution

Aucune.

Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

Résultat obtenu

Cette procédure génère un fichier mouvement à destination de la procédure UPDT de la version 3.0 ainsi qu'un fichier des mouvements rejetés.

MB25 - Description des étapes

Reprise du fichier MB : PTU926

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7AR	Rép. base : AR	Entrée	Données de la Base de développement
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7MB	Entrée utilisateur	Entrée	Mouvements MB 2.5
PAC7MV	Rép. user : MB25MV	Sortie	Mouvements de reprise pour UPDT
PAC7ME	Rép. user : MB25ME	Sortie	Mouvements rejetés
PAC7EF	Rép. user : MB25EF926	Etat	Compte-rendu de reprise
PAC7DD	Rép. user : MB25DD926	Etat	Contrôle autorisation

MB25 - Script d'exécution

```
REM * -----  
REM *      VISUALAGE PACBASE  
REM *  
REM * -----  
REM *      RETRIEVAL OF MB FILE  
REM *  
REM * -----  
REM *  
<job id=MB25>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "MB25"  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>  
  
<script language="VBScript">  
  
If c_error = 1 then Wscript.Quit (1) End If  
  
Call Msg_Log (Array("1022" , "PTU926"))  
'-----  
WshEnv("PAC7AE") = Rep_SKEL & "\AE"  
WshEnv("PAC7AR") = Rep_BASE & "\AR"
```

```

Call BvpEnv("PTU926","PAC7MB",RepT_USR & "\MB25")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU926","PAC7EF",Rep_USR & "\MB25EF926.txt")
Call BvpEnv("PTU926","PAC7DD",Rep_USR & "\MB25DD926.txt")
Call BvpEnv("PTU926","PAC7ME",Rep_USR & "\MB25ME.txt")
Call BvpEnv("PTU926","PAC7MV",Rep_USR & "\MB25MV.txt")
Return = WshShell.Run("BVPTU926.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU926")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Reprise mouvements GY (GY25)

GY25 - Présentation générale

Principe

Cette procédure permet de reprendre des mouvements GY de la version 2.0 ou 2.5 vers la version 3.0.

Condition d'exécution

Aucune.

Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

Résultat obtenu

Cette procédure génère un fichier mouvement à destination de la procédure UPDP de la version 3.0 ainsi qu'un fichier des mouvements rejetés.

GY25 - Description des étapes

Reprise du fichier GY : PTU927

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7AR	Rép. base : AR	Entrée	Données de la Base de développement
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7GY	Entrée utilisateur	Entrée	Mouvements GY 2.5
PAC7MV	Rép. user : GY25MV	Sortie	Mouvements de reprise pour UPDP
PAC7ME	Rép. user : GY25ME	Sortie	Mouvements rejetés
PAC7EF	Rép. user : GY25EF927	Etat	Compte-rendu de reprise
PAC7DD	Rép. user : GY25DD927	Etat	Contrôle autorisation

GY25 - Script d'exécution

```
REM * -----  
REM *      VISUALAGE PACBASE  
REM *  
REM * -----  
REM *      RETRIEVAL OF GY FILE  
REM *  
REM * -----  
REM *  
<job id=GY25>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "GY25"  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>  
  
<script language="VBScript">  
  
If c_error = 1 then Wscript.Quit (1) End If  
  
Call Msg_Log (Array("1022" , "PTU927"))  
'-----  
WshEnv("PAC7AE") = Rep_SKEL & "\AE"  
WshEnv("PAC7AR") = Rep_BASE & "\AR"
```

```

Call BvpEnv("PTU927","PAC7GY",RepT_USR & "\GY25")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU927","PAC7EF",Rep_USR & "\GY25EF927.txt")
Call BvpEnv("PTU927","PAC7DD",Rep_USR & "\GY25DD927.txt")
Call BvpEnv("PTU927","PAC7ME",Rep_USR & "\GY25ME.txt")
Call BvpEnv("PTU927","PAC7MV",Rep_USR & "\GY25MV.txt")
Return = WshShell.Run("BVPTU927.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU927")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Reprise mouvements MB (MB30)

MB30 - Présentation générale

Principe

Cette procédure permet de reprendre des mouvements MB de la version 3.0 vers la version 2.0 ou 2.5.

Condition d'exécution

Aucune.

Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

Résultat obtenu

Cette procédure génère un fichier mouvement à destination de la procédure UPDT de la version 2.0 ou 2.5 ainsi qu'un fichier des mouvements rejetés.

MB30 - Description des étapes

Reprise du fichier MB : PTU928

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7AR	Rép. base : AR	Entrée	Données de la Base de développement
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7MB	Entrée utilisateur	Entrée	Mouvements MB 3.0
PAC7MV	Rép. user : MB30MV	Sortie	Mouvements de reprise pour UPDT
PAC7ME	Rép. user : MB30ME	Sortie	Mouvements rejetés
PAC7EF	Rép. user : MB30EF928	Etat	Compte-rendu de reprise
PAC7DD	Rép. user : MB30DD928	Etat	Contrôle autorisation

MB30 - Script d'exécution

```
REM * -----  
REM *      VISUALAGE PACBASE  
REM *  
REM * -----  
REM *      RETRIEVAL OF MB FILE  
REM *  
REM * -----  
REM *  
<job id=MB30>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "MB30"  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>  
  
<script language="VBScript">  
  
If c_error = 1 then Wscript.Quit (1) End If  
  
Call Msg_Log (Array("1022" , "PTU928"))  
'-----  
WshEnv("PAC7AE") = Rep_SKEL & "\AE"  
WshEnv("PAC7AR") = Rep_BASE & "\AR"
```



```

Call BvpEnv("PTU928","PAC7MB",RepT_USR & "\MB30")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU928","PAC7EF",Rep_USR & "\MB30EF928.txt")
Call BvpEnv("PTU928","PAC7DD",Rep_USR & "\MB30DD928.txt")
Call BvpEnv("PTU928","PAC7ME",Rep_USR & "\MB30ME.txt")
Call BvpEnv("PTU928","PAC7MV",Rep_USR & "\MB30MV.txt")
Return = WshShell.Run("BVPTU928.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU928")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Reprise mouvements GY (GY30)

GY30 - Présentation générale

Principe

Cette procédure permet de reprendre des mouvements GY de la version 3.0 vers la version 2.0 ou 2.5.

Condition d'exécution

Aucune.

Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

Résultat obtenu

Cette procédure génère un fichier mouvement à destination de la procédure UPDP de la version 2.0 ou 2.5 ainsi qu'un fichier des mouvements rejetés.

GY30 - Description des étapes

Reprise du fichier GY : PTU929

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7AR	Rép. base : AR	Entrée	Données de la Base de développement
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7GY	Entrée utilisateur	Entrée	Mouvements GY 3.0
PAC7MV	Rép. user : GY30MV	Sortie	Mouvements de reprise pour UPDP
PAC7ME	Rép. user : GY30ME	Sortie	Mouvements rejetés
PAC7EF	Rép. user : GY30EF929	Etat	Compte-rendu de reprise
PAC7DD	Rép. user : GY30DD929	Etat	Contrôle autorisation

GY30 - Script d'exécution

```
REM * -----  
REM *      VISUALAGE PACBASE  
REM *  
REM * -----  
REM *      RETRIEVAL OF GY FILE  
REM *  
REM * -----  
<job id=GY30>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "GY30"  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>  
  
<script language="VBScript">  
  
If c_error = 1 then Wscript.Quit (1) End If  
  
Call Msg_Log (Array("1022" , "PTU929"))  
'-----  
WshEnv("PAC7AE") = Rep_SKEL & "\AE"  
WshEnv("PAC7AR") = Rep_BASE & "\AR"
```

```

Call BvpEnv("PTU929","PAC7GY",RepT_USR & "\GY30")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU929","PAC7EF",Rep_USR & "\GY30EF929.txt")
Call BvpEnv("PTU929","PAC7DD",Rep_USR & "\GY30DD929.txt")
Call BvpEnv("PTU929","PAC7ME",Rep_USR & "\GY30ME.txt")
Call BvpEnv("PTU929","PAC7MV",Rep_USR & "\GY30MV.txt")
Return = WshShell.Run("BVPTU929.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU929")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Reprise mouvements PJ (PJ25)

PJ25 - Présentation générale

Principe

Cette procédure (PJ25) permet de reprendre le fichier PJ, fichier journal (fichier séquentiel des mouvements), en un nouveau fichier d'archive au format 3.0.

Condition d'exécution

Aucune.

Edition obtenue

Cette procédure édite un compte-rendu indiquant le nombre de mouvements lus soit le nombre de mouvements repris tels quels, c'est à dire dans leur format 2.n, et le nombre de mouvements transformés en en mouvements 3.0; et le nombre d'enregistrements écrits. Attention, le nombre d'enregistrements écrits peut être très supérieur au nombre d'enregistrements lus. A partir d'un an- cien mouvement 2.n, plusieurs mouvements 3.0 peuvent être créés notamment pour les méta-entités et les entités utili- sateurs.

Résultat obtenu

Un fichier journal PJ au nouveau format 3.0.

Remarques

Cette opération de transformation du journal est facultative. Elle doit être exécutée si des procédures batch le nécessitent (Utilisation de Pac/Transfer).

Cette reprise s'effectue exclusivement des versions 2.0 ou 2.5 vers la version 3.0.

La reprise de certains mouvements nécessite la recherche d'informations dans une base 3.0. Il est fort possible que les données correspondantes n'existent plus dans la nouvelle base (session ou bibliothèque supprimée par exemple). Dans ce cas, l'ancien mouvement sera repris dans son format 2.n.

PJ25 - Description des étapes

Traitement du fichier séquentiel des mouvements : PTU918

Code	Nom physique	Type	Libellé
PAC7PJ	Rép. save : PJ25	Entrée	Fichier journal ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7AR	Rép. base : AR	Entrée	Données de la Base de développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de développement
PAC7AY	Rép. base : AY	Entrée	Données extensions de la base de développement
PAC7JP	Rép. save : PJ	Sortie	Fichier journal au format 3.0 (longueur=170)
PAC7ET	Rép. user : PJ25ET918	Etat	Compte-rendu

PJ25 - Script d'exécution

```
REM * -----  
REM *      VISUALAGE PACBASE  
REM *  
REM * -----  
REM *      RETRIEVAL OF PJ FILE  
REM *  
REM * -----  
REM *  
<job id=PJ25>  
  
<script language="VBScript">  
Dim MyProc
```

```

MyProc = "PJ25"
</script>

<script language="VBScript" src="INIT.vbs"/>
<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTU918"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
Call BvpEnv("PTU918","PAC7PJ",Rep_SAVE & "\PJ25")
Call BvpEnv("PTU918","PAC7ET",Rep_SAVE & "\PJ25ET918.txt")
Call BvpEnv("PTU918","PAC7JP",Rep_SAVE & "\PJ")
Return = WshShell.Run("BVPTU918.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU918")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Procédures - Tableau récapitulatif des changements

Liste des nouvelles procédures

Procédure	Remarques
ARCH	Archivage de la Base administration
INAE	Initialisation du fichier des libellés d'erreur (AE)
INGU	Initialisation du fichier des codes utilisateur (GU)
REOR	Réorganisation de la Base administration
REST	Restauration de la Base administration
PACS	Sauvegarde de la Base administration (option SAVE)
VINS	Mise à jour du Modèle administration
PACS	Gestion de la Base de développement

Procédure	Remarques
GY25	Reprise d'un fichier GY pour UPDP
MB25	Reprise d'un fichier MB pour UPDT
GY30	Reprise d'un fichier GY pour UPDP vers 2.5
MB30	Reprise d'un fichier MB pour UPDT vers 2.5
PC25	Reprise de la Base de développement
PE25	Reprise des paramètres utilisateur (PE)
PG20	Reprise des commandes édition-génération 2.0 (PG)
PG25	Reprise des commandes édition-génération 2.5 (PG)
PJ25	Reprise du journal 2.5
UV25	Reprise des paramètres Pac/Transfer (UV)
PP25	Reprise de l'environnement PEI (PP)
CHPM	Compte-rendu contrôle environnement et éléments de de la base
CPPM	Comparaison base avec configuration utilisateur
EXPM	Extraction des environnements
HIPM	Mise en production des éléments
SIPM	Simulation de génération
TRED	Pac/Transfer : édition paramètres

Liste des procédures supprimées depuis la release 2.5

Procédure	Programmes	Remarques
CPSN	PTU850 PTU855	Intégrée dans procédure PACX
EMSN	PTU810	
MESN	PTU815	
MLIB	PTU100 PTU120	Intégrée dans procédure PACS
SASN	PTU130 PTU140	Intégrée dans procédure PACS
SAVE	PTU500	Intégrée dans procédure PACS
UXSR	UTIXSR	Intégrée dans procédure PACS
CRYP	PACU99	
PARM	PACU15 PACU80	
LOAE	PACU80	
REAG	PTU560	
SVAG	PTU550	

Procédure	Programmes	Remarques
GET0	PACTI1	
GET1	PACT41	
GET2	PACT41 PACT51	
GRPE	PACR40	
INPE	PACR01	
PP16	PACR90	
PRPE	PACR10	
RSPE	PACR61	
SVPE	PACR60	
RVDE	PREI00 PRE986	
RVKE	PREI40 PREI50	
TRRT	REUV802 PTUG90	
VDWN	PVA100 PVA110	
VPUR	PVA400	
VPU1	PVA300 PVA305 PVA310	
VPU2	PVA320	
LVBL	PTULVB	
QREO	PTUN00 PTUN10 PTUN40	
RPPG	PTU908	
RPTD	PTAR20	

Chapitre 8. Les composants

Composants de la partie Serveur

Présentation générale

D'un point de vue technique, le produit gère des données permanentes en mode conversationnel ou en mode batch grâce à l'utilisation de deux types de ressources :

- Des répertoires dans lesquels sont stockés les programmes constitutifs du système, ainsi que les paramètres nécessaires à son fonctionnement.
- Des fichiers permanents qui matérialisent les données manipulées par les programmes définis précédemment. Ces fichiers peuvent être classés en deux catégories :
 - Les fichiers système, qui ne sont pas liés à une Base de développement particulière et restent relativement stables.
 - Les fichiers évolutifs qui correspondent à une Base de développement et dont le volume varie en fonction des mises à jour effectuées.

Documentation en ligne

En plus des bibliothèques décrites dans les sous-chapitres précédents, le produit comprend le fichier contenant les libellés d'erreur et la documentation en ligne : AE.

Squelettes de génération

Le produit comprend par ailleurs les fichiers suivants :

- Un fichier squelette de génération SC, utilisé par le générateur Batch.
- Un fichier squelette de génération SG, utilisé par les générateurs Dialogue et Base de données.
- Un fichier squelette de génération SN, utilisé par le générateur eBusiness.
- Un fichier squelette de génération SR, utilisé par le générateur Reverse.
- Un fichier squelette de génération SP, utilisé par le générateur d'extracteurs PAF.
- Un fichier squelette SF, utilisé par le générateur d'extracteurs PAF.
- Un fichier squelette de génération SS, utilisé par le générateur eBusiness.

Base administration

Fichiers de la Base administration

- Le fichier des données AR.
- Le fichier des données extension AY.

- Le fichier des index AN.
- Le fichier journal AJ.
- Le fichier utilisateur GU.

Sauvegarde de la Base administration

La sauvegarde des fichiers de la Base administration est constituée de deux fichiers séquentiels à génération.

- La sauvegarde de la base (PE).

C'est un fichier de sauvegarde séquentielle des constituants de la Base administration : index (AN), données (AR) et extension (AY).

- La sauvegarde du journal (PJ).

Son objectif est d'accumuler tous les mouvements de mise à jour de la Base administration depuis son installation et ayant transité par le fichier des mouvements (AJ).

Au cas où sa taille deviendrait incompatible avec les contraintes d'exploitation, la procédure ARCH de la Base Administration permet de le diviser en plusieurs fichiers dont seul le plus récent est manipulé régulièrement.

Base de développement

Fichiers de la Base de développement

Ces fichiers contiennent toutes les données relatives au développement des applications.

- Le fichier des données AR.
- Le fichier des données extension AY.
- Le fichier des index AN.
- Le fichier journal AJ.

Tous les mouvements passés sur la base, que ce soit en batch ou en TP, sont conservés afin de permettre une restauration de la base en cas de défaillance des sécurités standard du système, et d'autre part de fournir la source d'informations à l'outil d'analyse de l'évolution de la base.

Ces mouvements sont normalement stockés dans le fichier de sauvegarde des mouvements (PJ), le fichier des mouvements ne servant que de transit entre le moment où le système les traite et le moment où la procédure ARCH les conduit sur leur support définitif.

Fichiers de sauvegarde Base de développement

Suivant l'organisation prise lors de la restauration, la sauvegarde du réseau est constituée de deux fichiers séquentiels à "rotation" (PC et PJ) ou de quatre fichiers séquentiels à "rotation" (PC PD PY et PJ).

- La sauvegarde de la Base ou des données (PC).

C'est un fichier de sauvegarde séquentielle des constituants de la Base de développement (données (AR), index (AN) et extension (AY) parfois).

- La sauvegarde des index de la Base de développement PD.
- La sauvegarde des données extension de la Base de développement PY.
- La sauvegarde du journal (PJ).

Son objectif est d'accumuler tous les mouvements de mise à jour de la Base de développement depuis son installation et ayant transité par le fichier des mouvements (AJ).

Au cas où sa taille deviendrait incompatible avec les contraintes d'exploitation, la procédure ARCH permet de le diviser en plusieurs fichiers dont seul le plus récent est manipulé régulièrement.

Modules - fichiers spécifiques

Pac/Impact :

- Fichier des critères déjà impactés (FQ).
- Fichier critères de recherche ou points d'entrée (FH).
- Fichier réduit des critères pour épuration (FR).
- Fichier résultats d'impact (FO).
- Fichier des entités en production (FP).

DSMS : Pour les utilisateurs disposant du module DSMS (voir également le manuel d'exploitation de ce module), un fichier de ce Module est consulté en batch et en TP par Base de développement.

Ce fichier contient la liste des entités concernées par chaque amélioration. Le numéro d'amélioration est renseigné par l'utilisateur lors de sa connexion à la base.

- Le fichier DSMS des éléments de la Base de développement (DC).

L'allocation et l'initialisation de ce fichier sont effectuées lors de l'installation du module DSMS.

La définition fournie à l'installation du produit doit être utilisée si le module DSMS n'a pas encore été installé sur le site.

PAF :

- Un fichier de travail PA pour PAF-TP et PUF-TP.

Un fichier indexé de travail est nécessaire à l'utilisation des modules PAF et PUF pour l'ensemble des programmes TP utilisateur accédant aux bases de même radical.

- Un fichier de travail pour PAF batch.

Un fichier indexé de travail est nécessaire à l'utilisation du module PAF par les programmes batch utilisateur. Ce fichier est alloué pour l'exécution et détruit en fin de job.

Extension module PAF

- Un fichier Schémas d'Extraction GS contenant les extracteurs de l'utilisateur et ses macro-commandes.

Bibliothèques et fichiers complémentaires

Les fichiers compléments du dictionnaire sont livrés automatiquement dans le répertoire \SYS\SKEL.

Module PQC :

- les fichiers BVPQCRA et BVPQCRF contiennent les règles de qualité standard.

- Membre spécifique du module Pacbench Quality Control

Fichier	Contenu ou format	Remarques
BVPQCRA	Fichier séquentiel anglais	Règles standard
BVPQCRF	Fichier séquentiel français	Règles standard

Chapitre 9. Annexes

Installation du Modèle de la Base administration

VINS pour Administration - Présentation générale

La procédure VINS effectue la mise à jour batch de la Base administration à partir de mouvements livrés par IBM.

Condition d'exécution

L'accès au conversationnel doit être fermé.

Anomalies d'exécution

Se reporter au chapitre "Les procédures batch de l'administrateur" sous-chapitre "Anomalies d'exécution".

Lorsque l'anomalie survient pendant l'exécution des programmes BVPACI30 ou BVPACI40, la base est laissée dans un état incohérent. Il convient alors, après avoir remédié au problème, de recharger la base avec récupération des mouvements archivés, puis d'exécuter à nouveau la procédure VINS.

VINS - Entrées / Traitements / Résultats

Cette procédure nécessite deux entrées utilisateur :

- une ligne contenant l'identification de l'utilisateur ainsi que le traitement à effectuer,
- les mouvements permettant de créer les méta-entités IBM et de reprendre les entités utilisateur clientes au format 'extension' : en aucun cas, l'utilisateur ne doit modifier le contenu de ces mouvements.

La structure de la ligne est la suivante :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code carte
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	'***'	Code bibliothèque
29	4	'VINS'	
33	1	'I'	Installation des méta-entités IBM

Editions obtenues

La procédure édite :

- un compte-rendu avec la liste des programmes exécutés,
- la liste des demandes avec indication des erreurs éventuelles,
- un compte-rendu des mises à jour effectuées par l'installation.

Résultat obtenu

Une fois la mise à jour effectuée, le réseau est prêt à être manipulé en conversationnel ou en mode batch.

Remarque importante

Il faut prévoir d'écrire les mouvements extraits pour la procédure REOR dans un fichier catalogué en prenant en compte la ligne indiquée en commentaire.

VINS - Description des étapes

Mise à jour de la Base administration : VINS

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7AJ	Rép. journal : AJ	Entrée	Journal de la Base administration
PAC7AN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PAC7AR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PAC7AY	Base Admin - Rép. base : AY	Entrée	Extension de la Base administration
PAC7MA	Système - Rép. skel : BVPMETAD	Entrée	Mouvements méta-entités IBM
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7BM	Rép. tmp : WBM	Entrée/sortie	Entrées utilisateur

Code	Nom physique	Type	Libellé
PAC7WD	Rép. tmp : WWD	Entrée/sortie	Mouvements extraits
PAC7ES	Rép. tmp : WES	Entrée/sortie	Mouvements extraits
PAC7TD	Rép. tmp : WTD	Entrée/sortie	Mouvements extraits
PAC7MR	Rép. tmp : WMR	Sortie	Mouvements extraits pour REOR sur Base Administration
PAC7MX	nul	Sortie	Entités non extraites
PAC7IA	Rép. user : VINGIA	Etat	Edition générale de l'enchaînement des programmes
PAC7EE	Rép. user : VINGEE	Etat	Compte-rendu
PAC7EQ	Rép. user : VINGEQ	Etat	Compte-rendu
PAC7EU	Rép. user : VINGEU	Etat	Compte-rendu
PAC7ER	Rép. user : VINGER	Etat	Compte-rendu
PAC7EZ	Rép. user : VINGEZ	Etat	Compte-rendu
PAC7DD	Rép. user : VINGDD	Etat	Compte-rendu
PAC7IE	Rép. user : VINGIE	Etat	Compte-rendu
PAC7IF	Rép. user : VINGIF	Etat	Compte-rendu
PAC7IG	Rép. user : VINGIG	Etat	Compte-rendu
PAC7IH	Rép. user : VINGIH	Etat	Compte-rendu

Codes retour :

- 0 : Pas d'erreur détectée
- 4 : Warning
- 8 : Pas d'autorisation d'accès procédure batch ou base invalide (dans ce cas, relancer la procédure en indiquant 'D' en colonne 16)
- 12 : Erreur d'entrée-sortie sur un fichier

VINS - Script d'exécution

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      - DICTIONARY UPDATING WITH IBM MODEL DEVPT -
REM *
REM * -----
REM *
REM * THE VINS PROCEDURE PERFORMS A BATCH UPDATE OF THE
REM * DATABASE, BASED ON TRANSACTIONS PROVIDED.
REM *
REM * INPUT :
REM * - USER IDENTIFICATION LINE (REQUIRED)

```

```

REM *      COL 2 : '*'
REM *      COL 3 : USERIDXX
REM *      COL 11 : PASSWORD
REM *      COL 27 : ' ' - NO UPDATE
REM *          'S' - UPDATE SIMULATION WITH PRINTING OF
REM *                  OF LIST OF U.E.'S TO BE CANCELLED
REM *          'F' - FORCING THE CANCELLATION OF U.E.'S
REM *                  WITH THE SAME CODES IN LOWER
REM *                  LEVEL LIBRARIES
REM * -----
REM *
<job id=VINS>

```

```

<script language="VBScript">
Dim MyProc
MyProc = "VINS"
</script>

```

```

<script language="VBScript" src="INIT.vbs"/>

```

```

<script language="VBScript">

```

```

If c_error = 1 then Wscript.Quit (1) End If

```

```

Call Msg_Log (Array("1029" ))
'-----
Call StateList (base, statusL)

```

```

If c_error = 1 then Wscript.Quit (1) End If

```

```

Call Msg_Log (Array("1022" , "VINS"))
'-----

```

```

WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
Call BvpEnv("VINS", "PAC7BM", Rep_TMP & "\WBM.tmp")
Call BvpEnv("VINS", "PAC7ES", Rep_TMP & "\WES.tmp")
Call BvpEnv("VINS", "PAC7DD", Rep_USR & "\VINSDD.txt")
WshEnv("PAC7EE") = "NUL"
Call BvpEnv("VINS", "PAC7EQ", Rep_USR & "\VINSEQ.txt")
Call BvpEnv("VINS", "PAC7ER", Rep_USR & "\VINSEI.txt")
Call BvpEnv("VINS", "PAC7EU", Rep_USR & "\VINSEU.txt")
Call BvpEnv("VINS", "PAC7EZ", Rep_USR & "\VINSEZ.txt")
Call BvpEnv("VINS", "PAC7IA", Rep_USR & "\VINSIA.txt")
Call BvpEnv("VINS", "PAC7IE", Rep_USR & "\VINSIE.txt")
Call BvpEnv("VINS", "PAC7IF", Rep_USR & "\VINSIF.txt")
Call BvpEnv("VINS", "PAC7IG", Rep_USR & "\VINSIG.txt")
Call BvpEnv("VINS", "PAC7IH", Rep_USR & "\VINSIH.txt")

```



```

If base = "ADMIN" Then
WshEnv("PAC7MA") = Rep_SKEL & "\BVPMETAD"
Else
WshEnv("PAC7MA") = Rep_SKEL & "\BVPMETBA"
End If

'Example of Input File extracted from PACX/CPSN :
' Call BvpEnv("VINS","PAC7xx",RepT_USR & "\PACXxx.txt")
'With RepT_USR is Global User Directory.
Call BvpEnv("VINS","PAC7TD",Rep_USR & "\TD.txt")

WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("VINS","PAC7MR",Rep_TMP & "\WMR.tmp")
WshEnv("PAC7MX") = "NUL"
Call BvpEnv("VINS","PAC7WD",Rep_TMP & "\WWD.tmp")
Call BvpEnv("VINS","SYSEXT",Rep_TMP & "\VINSSYS.tmp")
Call RunCmdLog ("BVPVINS")
WshVolEnv("RC") = Return
Call Err_Cod(Return , 0 , "VINS")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
WshVolEnv("RC") = Return
Wscript.Quit (Return)

</script>
</job>

```

Installation du Modèle de la Base de développement

VINS - Présentation générale

La procédure VINS effectue la mise à jour batch du réseau à partir de mouvements livrés par IBM.

Les entités sont créées en inter-bibliothèque et en session 0001Z afin d'en permettre l'accès depuis n'importe quelle bibliothèque du réseau et à partir de n'importe quelle session.

Condition d'exécution

L'accès au conversationnel doit être fermé.

Anomalies d'exécution

Se reporter au chapitre "Les procédures batch de l'administrateur" sous-chapitre "Anomalies d'exécution".

Lorsque l'anomalie survient pendant l'exécution des programmes BVPACI30 ou BVPACI40, la base est laissée dans un état incohérent. Il convient alors, après avoir remédié au problème, de recharger la base avec récupération des mouvements archivés, puis d'exécuter à nouveau la procédure VINS.

VINS - Entrées / Traitements / Résultats

Cette procédure nécessite deux entrées utilisateur :

- une ligne contenant l'identification de l'utilisateur ainsi que le traitement à effectuer,
- les mouvements permettant de créer les méta-entités IBM et de reprendre les entités utilisateur clientes au format 'extension' : en aucun cas, l'utilisateur ne doit modifier le contenu de ces mouvements.

La structure de la ligne est la suivante :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code carte
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	'***'	Code bibliothèque
29	4	'VINS'	
33	1	'I'	Installation des méta-entités IBM
		'R'	Reprise des entités utilisateur au format 'extension'
		' '	'I' + 'R'

Editions obtenues

La procédure éditée :

- un compte-rendu avec la liste des programmes exécutés,
- la liste des demandes avec indication des erreurs éventuelles,
- un compte-rendu des mises à jour effectuées par l'installation,
- un compte-rendu des mises à jour effectuées par la reprise.

Résultat obtenu

Une fois la mise à jour effectuée, le réseau est prêt à être manipulé en conversationnel ou en mode batch.

Pour la reprise, un fichier séquentiel de mouvements d'épuration destinés à la procédure de réorganisation est généré. Suite à la reprise, la réorganisation de la base est obligatoire.

Remarque importante

Il faut prévoir d'écrire les mouvements extraits pour la procédure REOR dans un fichier catalogué.

VINS - Description des étapes

Description des Etapes

Mise à jour du réseau : VINS

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base administration
PAC7AJ	Rép. journal : AJ	Entrée	Journal de la Base de développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de développement
PAC7AR	Rép. base : AR	Entrée	Données de la Base de développement
PAC7AY	Rép. base : AY	Entrée	Extension de la Base de développement
PAC7MA	Système - Rép. skel : BVPMETBA	Entrée	Mouvements méta-entités IBM
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7BM	Rép. tmp : WBM	Entrée/sortie	Entrées utilisateur
PAC7WD	Rép. tmp : WWD	Entrée/sortie	Mouvements extraits
PAC7ES	Rép. tmp : WES	Entrée/sortie	Mouvements extraits
PAC7TD	Rép. tmp : WTD	Entrée/sortie	Mouvements extraits
PAC7MR	Rép. tmp : WMR	Sortie	Mouvements extraits pour REOR

Code	Nom physique	Type	Libellé
PAC7MX	nul	Sortie	Entités non extraites
PAC7IA	Rép. user : VINSIA	Etat	Edition générale de l'enchaînement des programmes
PAC7EE	Rép. user : VINSEE	Etat	Compte-rendu
PAC7EQ	Rép. user : VINSEQ	Etat	Compte-rendu
PAC7EU	Rép. user : VINSEU	Etat	Compte-rendu
PAC7ER	Rép. user : VINSER	Etat	Compte-rendu
PAC7EZ	Rép. user : VINSEZ	Etat	Compte-rendu
PAC7DD	Rép. user : VINSDD	Etat	Compte-rendu
PAC7IE	Rép. user : VINSIE	Etat	Compte-rendu
PAC7IF	Rép. user : VINSIF	Etat	Compte-rendu
PAC7IG	Rép. user : VINSIG	Etat	Compte-rendu
PAC7IH	Rép. user : VINSIH	Etat	Compte-rendu

Codes retour :

- 0 : Pas d'erreur détectée sur les fichiers
- 4 : Enregistrement du fichier journal erroné (Date ou numéro de session non numérique)
- 8 : Pas d'autorisation d'accès procédure batch ou base invalide (dans ce cas, relancer la procédure en indiquant 'D' en colonne 16)
- 12 : Erreur d'entrée-sortie sur un fichier

VINS - Script d'exécution

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      - DICTIONARY UPDATING WITH IBM MODEL DEVPT -
REM *
REM * -----
REM *
REM * THE VINS PROCEDURE PERFORMS A BATCH UPDATE OF THE
REM * DATABASE, BASED ON TRANSACTIONS PROVIDED.
REM *
REM * INPUT :
REM * - USER IDENTIFICATION LINE (REQUIRED)
REM *   COL 2 : '*'
REM *   COL 3 : USERIDXX
REM *   COL 11 : PASSWORD
REM *   COL 27 : ' ' - NO UPDATE
REM *           'S' - UPDATE SIMULATION WITH PRINTING OF
REM *           OF LIST OF U.E.'S TO BE CANCELLED

```

```

REM *          'F' - FORCING THE CANCELLATION OF U.E.'S
REM *          WITH THE SAME CODES IN LOWER
REM *          LEVEL LIBRARIES
REM * -----
REM *
<job id=VINS>

<script language="VBScript">
Dim MyProc
MyProc = "VINS"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1029" ))
'-----
Call StateList (base, statusL)

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "VINS"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
Call BvpEnv("VINS", "PAC7BM", Rep_TMP & "\WBM.tmp")
Call BvpEnv("VINS", "PAC7ES", Rep_TMP & "\WES.tmp")
Call BvpEnv("VINS", "PAC7DD", Rep_USR & "\VINSDD.txt")
WshEnv("PAC7EE") = "NUL"
Call BvpEnv("VINS", "PAC7EQ", Rep_USR & "\VINSEQ.txt")
Call BvpEnv("VINS", "PAC7ER", Rep_USR & "\VINSER.txt")
Call BvpEnv("VINS", "PAC7EU", Rep_USR & "\VINSEU.txt")
Call BvpEnv("VINS", "PAC7EZ", Rep_USR & "\VINSEZ.txt")
Call BvpEnv("VINS", "PAC7IA", Rep_USR & "\VINSIA.txt")
Call BvpEnv("VINS", "PAC7IE", Rep_USR & "\VINSIE.txt")
Call BvpEnv("VINS", "PAC7IF", Rep_USR & "\VINSIF.txt")
Call BvpEnv("VINS", "PAC7IG", Rep_USR & "\VINSIG.txt")
Call BvpEnv("VINS", "PAC7IH", Rep_USR & "\VINSIH.txt")

If base = "ADMIN" Then
WshEnv("PAC7MA") = Rep_SKEL & "\BVPMETAD"
Else
WshEnv("PAC7MA") = Rep_SKEL & "\BVPMETBA"

```

```

End If

'Example of Input File extracted from PACX/CPSN :
' Call BvpEnv("VINS","PAC7xx",RepT_USR & "\PACXxx.txt")
'With RepT_USR is Global User Directory.
Call BvpEnv("VINS","PAC7TD",Rep_USR & "\TD.txt")

WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("VINS","PAC7MR",Rep_TMP & "\WMR.tmp")
WshEnv("PAC7MX") = "NUL"
Call BvpEnv("VINS","PAC7WD",Rep_TMP & "\WWD.tmp")
Call BvpEnv("VINS","SYSEXT",Rep_TMP & "\VINSSYS.tmp")
Call RunCmdLog ("BVPVINS")
WshVolEnv("RC") = Return
Call Err_Cod(Return , 0 , "VINS")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
WshVolEnv("RC") = Return
Wscript.Quit (Return)

</script>
</job>

```




Référence : DELNT001303F - 5957

Imprimé en France

(1P) P/N: DELNT001303F - 5957

